

# Equity-promoting Integer Programming Approaches For Medical Resident Rotation Scheduling

Shutian Li, Karmel S. Shehadeh, Frank E. Curtis, Beth R. Hochman

<sup>a</sup>*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA*

<sup>b</sup>*Department of Surgery, Columbia University Irving Medical Center, New York, NY, USA*

---

## Abstract

Motivated by our collaboration with a residency program at an academic health system, we propose new integer programming (IP) approaches for the *resident-to-rotation assignment problem* (RRAP). Given sets of residents, resident classes, and departments, as well as a block structure for each class, staffing needs, rotation requirements for each class, program rules, and resident vacation requests, the RRAP consists of finding a feasible yearlong rotation schedule specifying resident assignments to rotations and vacation times. We first present an IP formulation for the RRAP, which mimics the manual method for generating rotation schedules in practice and can be easily implemented and efficiently solved using off-the-shelf optimization software. However, it can lead to disparities in satisfying vacation requests among residents. To mitigate such disparities, we derive an equity-promoting counterpart, which finds an optimal rotation schedule that maximizes the number of satisfied vacation requests and minimizes a measure of disparity in satisfying vacation requests. Then, we propose a computationally efficient Pareto Search Algorithm capable of finding the complete set of Pareto optimal solutions to the equity-promoting IP within a time that is suitable for practical implementation. In addition, we present a user-friendly tool implementing the proposed models to automate the generation of the rotation schedule. Finally, we construct diverse RRAP instances based on data from our collaborator and conduct extensive experiments to illustrate the potential practical benefits of our proposed approaches. Our results demonstrate the computational efficiency and implementability of our approaches and underscore their potential to enhance fairness in resident rotation scheduling.

*Keywords:* Resident scheduling; Fairness; Optimization; Integer Programming

---

## 1. Introduction

Upon graduating from medical school, new doctors join a certified residency program to fulfill specialty board certification requirements as part of the pathway to independent practice. Residency program lengths and structures vary by specialty, but most programs in the United States (U.S.)

---

*Email address:* [kas720@lehigh.edu](mailto:kas720@lehigh.edu) (Karmel S. Shehadeh)

require around three to five years, commonly known as *post-graduate years* (PGY). For example, general surgery programs (including the one for Columbia University Irving Medical Center (CUIMC)) require five years. During each of these years, residents rotate through different hospital departments (services or divisions) to gain required clinical training and specialization under the supervision of more senior physicians and provide care to patients in these departments.

Medical schools administer residency programs in conjunction with their affiliated hospitals. The Accreditation Council for Graduate Medical Education (ACGME) sets the guidelines and expectations for residency programs in the U.S. Some of these guidelines apply to all specialties. Others are specialty-specific, such as the rotation period in each clinical setting. Residency programs use these guidelines to decide the list of rotations for each class of residents (PGY1–PGY5), then construct a yearlong *rotation schedule* that specifies the assignment of residents to *rotation* periods in different departments during the academic year. The schedule also specifies vacation time for each resident.

The exact approach for constructing the annual rotation schedule may vary between residency programs. However, most programs, including CUIMC, divide the academic year (e.g., 52–53 weeks at CUIMC) into time *blocks*, each consisting of several consecutive weeks within the academic year. The length of each block (i.e., the number of weeks) can be different, and some blocks may have more weeks than others. Assigning a resident to a block is equivalent to assigning this resident to a rotation period in a specific department. The length and structure of each rotation depend on the resident class, ACGME guidelines, residency program requirements, and staffing needs. Moreover, each resident must take a mandatory number of vacation weeks during the academic year that adhere to ACGME rules on rest periods. Residents often submit their requests for vacation time before the annual rotation schedule is constructed.

In most hospitals, the program director or chosen resident (e.g., the chief resident at CUIMC) manually constructs the annual rotation schedule for each class before the academic year starts. Constructing the annual rotation schedule is challenging, lengthy, and laborious for various reasons (Erhard et al., 2018; Guo et al., 2014; Lemay et al., 2017; Perelstein et al., 2016; Smalley and Keskinocak, 2016). First, the schedule must adhere to accreditation standards and satisfy different class-specific rotation and educational requirements. Second, it must fulfill multiple staffing needs of different departments and affiliated hospitals. Third, it should accommodate residents’ vacation preferences. Finally, the rotation schedule should ensure equity among residents (e.g., no resident receives a more desirable vacation assignment than any other resident for the academic year) while also considering all other requirements and logistics preferences of the residency program. Indeed, the chief resident spends several weeks manually designing a rotations schedule that often fails to fulfill these constraints simultaneously. In addition, manual methods often produce inequitable rotation schedules that are also challenging to adjust during the academic year. Mathematical formulations of the rotation scheduling problem are also challenging to solve.

These challenges motivate the need for computationally efficient and implementable optimization models that help automate rotation scheduling. The ACGME has also called for methods to generate better and equitable rotation schedules to help improve residents’ satisfaction and retention (Block et al., 2013; Colbert et al., 2017; Proano and Agarwal, 2018). However, as discussed in Section 2, rotation scheduling has received less attention than the shift scheduling problem. Moreover, existing formulations of the problem are challenging to solve, and few studies have attempted to address the issue of equity (fairness) among residents.

Motivated by these critical issues and our collaboration with CUIMC, this study develops, analyzes, and deploys new integer programming approaches for the *resident-to-rotation assignment problem* (RRAP). Given sets of residents, resident classes, and departments, as well as a block structure for each class, rotation requirements for each class, program rules and requirements, and resident vacation requests, the RRAP is a feasibility problem that consists of finding a feasible rotation schedule specifying resident assignments to rotations and vacations. We first derive an IP formulation for the RRAP, which finds a feasible rotation schedule that satisfies all rotation and residency program requirements. As we later show, such a blind, feasible assignment to rotations results in inequitable rotation schedules, specifically disparities in satisfying vacation requests among residents, with some being given vacations according to their preferences and others being given vacations in weeks different from their preferred weeks. To address this issue, we derive an *equity-promoting* counterpart that ensures equity in the number of vacation requests granted based on residents’ preferences. Satisfying vacation preferences has been considered a criterion for other scheduling endeavors (see, e.g., Becker et al. (1982); Lemay et al. (2016); Cohn et al. (2009); Shahraki et al. (2022); Gross et al. (2019); Uhde et al. (2020)), and its absence has been noted as a drawback that impacts resident satisfaction, among others. As highlighted by Howard et al. (2020), the challenge of accommodating vacation requests and equity in satisfying these requests has prompted the exploration of alternative, non-automated methods that lead to unfair schedules. This underscores the significance of improving equity in fulfilling vacation requests as an important indicator of schedule quality and residents’ satisfaction. Finally, we note that residents of the same class will be assigned the same workload in terms of required rotations throughout the academic year. So there, there is no disparity in this aspect.

### 1.1. Contributions

Our main contributions, both methodological and practical, can be summarized as follows.

#### 1. New IP Approaches for the RRAP.

- (a) We propose new IP models for the RRAP. These models adhere to ACGME guidelines and incorporate decisions and constraints related to class-specific rotation and program requirements, department staffing needs, vacation requirements, and other practical considerations. We first derive an IP formulation that finds a feasible rotation schedule that satisfies these constraints. This formulation mimics the manual method for generating ro-

tation schedules in practice and provides the foundation for the equity-neutral and equity-promoting models. The equity-neutral model seeks to find an optimal rotation schedule that maximizes the total number of satisfied vacation requests. The equity-promoting counterpart additionally incorporates an inequity measure in the objective to minimize disparities in satisfying vacation requests among residents.

- (b) The feasibility problem and equity-neutral model can be easily implemented and efficiently solved using off-the-shelf optimization software, enabling the implementation of the model in practice. In contrast, the proposed equity-promoting RRAP model has two conflicting objectives: maximize the total number of satisfied vacation requests and minimize a measure of inequity in satisfied requests among residents. Identifying the entire set of Pareto optimal (non-dominated) rotation schedules to the equity-promoting RRAP problem via traditional methods, such as the  $\varepsilon$ -constraint method, is computationally challenging. To address this challenge, we propose a computationally efficient *Pareto Search Algorithm* capable of finding the complete set of Pareto optimal solutions for large instances of the problem within a reasonable time suitable for practical implementation.
2. **User-Friendly Tool.** We developed a user-friendly tool implementing the proposed models to ensure the adaptation in practice and automate the annual rotation schedule generation.
  3. **Computational Results and Insights.** We construct diverse RRAP instances based on data from CUIMC and conduct extensive experiments to illustrate the potential practical benefits of our proposed approaches. Our results (a) demonstrate the computational efficiency and implementability of our approaches and underscore their potential to enhance fairness in resident rotation scheduling; (b) show how equity-neutral models lead to inequitable rotation schedules and disparities in satisfying vacation requests; (c) illustrate how different choices of the inequity measure in the equity-promoting model result in different sets of Pareto-optimal rotation schedules; and (d) emphasize the importance of integrating rotation and vacation scheduling decisions to ensure equity among residents and show the negative consequences on fairness when adopting a sequential approach that separates the rotation and vacation scheduling decisions. Although inspired by our collaborating residency program, our approaches are generic and can be adapted by other programs to automate rotation scheduling.

### 1.2. Structure of the Paper

The remainder of the paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we provide details about CUIMC’s general surgery residency program. In Section 4, we present our proposed models. We present our Pareto Search Algorithm in Section 5. In Section 6, we present the RRAP tool. We present our numerical experiments and corresponding insights in Section 7. Finally, we draw conclusions in Section 8.

## 2. Relevant Literature

Medical professionals (e.g., physicians, nurses, etc.) and personal scheduling problems have attracted much attention from the operations research community. These problems are challenging from a computational point of view, and each poses some specific modeling challenges. We refer to [Brucker et al. \(2011\)](#); [Van den Bergh et al. \(2013\)](#); [Abdalkareem et al. \(2021\)](#) for comprehensive surveys on these problems. We refer to [Erhard et al. \(2018\)](#) for a recent survey on physician scheduling, including resident scheduling. Two main scheduling problems arise in resident training—rotation scheduling and shift scheduling. Typically, the former problem must be solved first before the academic year starts to construct the rotation schedule, specifying residents’ assignments to specific departments in specific time blocks and vacation periods. The shift scheduling problem is then solved for each department to plan the daily and shift schedules for residents rotating in that department. Recent surveys indicate that the rotation scheduling problem received significantly less attention than the shift scheduling problem [Erhard et al. \(2018\)](#). In addition, limited research has considered the issue of improving equity among residents. Our research contributes to filling this gap. Next, we focus on discussing relevant studies on rotation scheduling.

[Franz and Miller \(1993\)](#) is one of the early works on rotation scheduling. They proposed an IP model for assigning residents to rotations over one year and solved it using a rounding heuristic. [Guo et al. \(2014\)](#) proved that the resident scheduling problem is NP-complete and developed a greedy heuristic to solve it. [Cire et al. \(2019\)](#) formulated the rotation scheduling problem using a network-flow model. Recently, [Guo et al. \(2023\)](#) proposed a two-stage partial fixing approach for solving the residency block scheduling problem. Note that merely assigning residents to rotations and vacations according to program requirements may result in inequitable rotation and vacation schedules and, consequently, residents’ dissatisfaction ([Colbert et al., 2017](#)). However, most studies, including those mentioned above, formulate the rotation scheduling problem as a feasibility problem or do not incorporate equity measures in their models. Next, we review the limited literature considering equity in resident rotation scheduling.

[Akbarzadeh and Maenhout \(2021\)](#) studied the problem of assigning medical students to specific disciplines and hospitals. Specifically, they considered a residency program where students could customize their training by selecting a preferred subset of the disciplines from an elective list. They proposed a mixed-integer program (MIP) to find feasible schedules satisfying all educational and hospital requirements while considering student preferences and ensuring equity in meeting them. [Akbarzadeh and Maenhout \(2021\)](#) solved their MIP using a customized branch and price algorithm. [Smalley and Keskinocak \(2016\)](#) discussed the issue of ensuring an equitable educational experience among residents during their rotation. They proposed an IP formulation to find a resident rotation schedule. The IP model ensures that residents of the same class are assigned to rotations in the same set of departments. [Proano and Agarwal \(2018\)](#) proposed a multi-objective IP model for rotation assignment.

Bard et al. (2017) proposed a MIP model to determine the annual block schedule of Internal Medicine residents that hierarchically minimizes the maximum deviation between the number of clinic sessions each resident must attend from the average; restricts the number of times residents are assigned to a night float block immediately before or after an intensive care unit block; and minimizes the maximum deviation between patients seen in clinic during any month over the year from the average. Bard et al. (2017) solved their MIP using a heuristic that yielded locally optimal alternative solutions. Castaño and Velasco (2020) focused on fairness from the perspective of hospitals and proposed an MIP formulation to find resident rotation schedules that minimize the variance in the number of residents assigned to each department. They developed a heuristic based on variable neighborhood searches to obtain near-optimal solutions.

We contribute to this literature with new and computationally efficient equity-promoting IP methodologies for rotation scheduling. Our work differs from the above pioneering studies in the following aspects. First, most studies assume that the academic year is divided into multiple periods or blocks of the same length, typically a month or a week (e.g., Bard et al. (2016, 2017); Guo et al. (2023); Smalley and Keskinocak (2016)). Such models can only be used for constructing rotation schedules with equal-length or similar rotations. In many residency programs, however, each class of residents may require a different training period in each department, i.e., the set of blocks and their length (and hence the number of departments and period of rotation in each) are different for each class. For example, at CUIMC, rotation schedules of PGY1 residents consist of blocks of 4 consecutive weeks in each required department, while those of PGY5 are comprised of blocks of 7-8 consecutive weeks in each required department (see Section 3). Our proposed models generalize existing models by accommodating block structure and length variations within resident classes. Second, most existing models do not require a resident to complete their rotation in each department before starting their rotation in another department. Our proposed model allows the resident to complete their rotation in each department before they start the next one (a requirement at CUIMC and other programs).

Third, although some studies have attempted to incorporate inequity measures in their models, each study adopted a specific measure. For example, Bard et al. (2016, 2017); Kraul (2020) used the maximum deviation from the mean as the inequity measure in their models, while Smalley and Keskinocak (2016) and Castaño and Velasco (2020) considered minimizing the difference between maximum and minimum outcomes. In contrast, our model allows for incorporating different inequity measures, and we compare rotation schedules constructed using a set of well-known measures. In addition, prior studies did not propose methodologies for identifying the trade-off between inequity and other objectives. In contrast, we develop a computationally efficient Pareto Search Algorithm that allows practitioners to analyze the trade-off between resident satisfaction and equity. Identifying the complete set of non-dominated rotation schedules in our context is beneficial as it enhances the decision-makers understanding of the trade-offs between the two objectives and

Table 1: An example of mandatory departments for each class of residents.

Department	PGY1	PGY2	PGY3	PGY4	PGY5
Hepatopancreaticobiliary/Endocrine Surgery (HPB)	✓		✓		✓
Colorectal Surgery (CR)	✓		✓		✓
Breast/Surgical Oncology (Breast)	✓		✓		
Vascular Surgery (Vascular)	✓		✓	✓	
Advanced Laparoscopic Surgery/Complex Hernia (Lap)	✓		✓		✓
Pediatric Surgery (Peds)	✓	✓			
Overlook Hospital (Overlook)	✓	✓	✓	✓	
Surgical Intensive Care Unit (SICU)	✓				
Thoracic Surgery (Thoracic)	✓		✓		
Night Float (Nights)	✓			✓	✓
Acute Care Surgery Consult (ACS-Consults)		✓			
Acute Care Surgery Operative (ACS-OR)		✓			
Allen Hospital Consult (Consults)		✓			
Cardiothoracic Intensive Care Unit (CTICU)		✓			
Renal Transplant (Renal)			✓		
Trauma–University Hospital (Trauma)			✓		
Acute Care Surgery (ACS)				✓	
Allen Hospital (Allen)				✓	
Elective (Elective)				✓	✓

enables them to select the most preferred schedule. Finally, most studies primarily address the assignment of residents to rotations with limited consideration for vacation scheduling. In contrast, we integrate rotation and vacation scheduling. In Section 7.3, we demonstrate the importance of integrating these decisions, especially to promote equity.

### 3. The General Surgery Residency Program at CUIMC

Our proposed models are partly based on the general surgery residency program at CUIMC. In this section, we provide details about this program to lay the foundation for the subsequent discussions. This program is a five-year clinical training program. Year 1 (PGY1) residents are recent medical graduates joining the program, while year 5 (PGY5) residents are the most senior. The program offers categorical and preliminary positions. A categorical position is a five-year-long training required for board certification. In contrast, a preliminary position is one to two years of training generally before entry into advanced specialty programs. Residents from other surgical specialties (e.g., cardiac surgery) also do rotations in this program; hence, their partial rotations within the general surgery departments must be considered. Clinical training occurs at four sites: Milstein Hospital, Allen Hospital, University Hospital (Newark), and Overlook Hospital.

Before the academic year starts, the chief resident collects staffing requirements from each department (e.g., the minimum and maximum number of residents required to serve in each department), residents’ information, and vacation requests. Then, s/he divides the academic year of each class into several blocks based on the number of residents, educational requirements, ACGME rules, departments’ needs, and other considerations. Since each class of residents may require a different length of training period in each department, the set of blocks and their length are different for each class. For example, in the 2022–2023 academic year (PGY1, PGY2, PGY3, PGY4, PGY5) rotations were divided into (4, 6, 6 to 7, 7 to 8, 8 to 9) weeks long blocks, respectively.

Each class has different requirements regarding the department they must rotate in (mandatory

	6/20	6/27	7/4	7/11	7/18	7/25	8/1	8/8	8/15	8/22	8/29	9/5	9/12	9/19	9/26	10/3	10/10		4/24	5/1	5/8	5/15	5/22	5/29	6/5	6/12	
<b>PGY5</b>	<b>Block 1 (8)</b>								<b>Block 2 (9)</b>								...	<b>Block 6 (8)</b>									
Resident A	Nights								v	HPB									Elective								
Resident B	v	CR									Lap								...	HPB							
<b>PGY4</b>	<b>Block 1 (7)</b>							<b>Block 2 (7)</b>							<b>Block 3 (7)</b>			...	<b>Block 7 (8)</b>								
Resident C	Overlook							v	Elective							v	Vascular			...	Allen						
Resident D	ACS			v	HPB				v	Nights			...	Vascular													
<b>PGY3</b>	<b>Block 1 (6)</b>					<b>Block 2 (6)</b>					<b>Block 3 (5)</b>				<b>Block 4 (6)</b>		...	<b>Block 9 (6)</b>									
Resident E	v	Renal TXP				Lap					CR				v	Trauma		...	v	Thoracic							
Resident F	Trauma					CR					v	Renal TXP				Overlook		...	Trauma								
<b>PGY2</b>	<b>Block 1 (7)</b>						<b>Block 2 (6)</b>						<b>Block 3 (6)</b>						...	<b>Block 8 (7)</b>							
Resident G	Consults						Allen						Consults						v	ACS-OR							
Resident H	ACS-OR						Peds Surgery						v	Allen						...	Consults						
<b>PGY1</b>	<b>Block 1 (5)</b>				<b>Block 2 (4)</b>				<b>Block 3 (5)</b>				<b>Block 4 (4)</b>				<b>Block 5 (5)</b>				...	<b>Block 12 (5)</b>					
Resident I	Allen				Vascular				Breast				v	VTF				Thoracic				...	Lap				
Resident J	VTF				Peds Surgery				SICU				Lap				v	CR				...	Overlook				

Figure 1: An example of rotation schedule of PGY1–PGY5 residents. There are ten residents in this example (two from each class). The number inside parentheses is the number of weeks in each block. Pink boxes marked with “v” indicate that a resident is on vacation that week.

departments). Table 1 provides an example of mandatory departments for each class. Categorical residents must attend the program during the academic year and rotate through all the mandatory departments for their classes. Generally, most residents rotate in each required department for their class once during the academic year. However, some classes may rotate more than once in some departments to satisfy their educational requirements or work time restrictions. For example, PGY2 needs to do service in the Acute Care Surgery (ACS) department twice (see Table 1). The first rotation is ACS-Consults, where residents learn clinical algorithms for common consults, including biliary disease, acute abdomen, bowel obstruction, and peripheral vascular disease. The second rotation is ACS-OR, which is more demanding. Specifically, in this rotation, residents perform all daytime operations on consults and elective patients in the Acute Care Service, manage the non-operative consult list, and rounds in the Surgical ICU on ACS inpatients.

The typical number of residents in this program often ranges from 50-65. In Figure 1, we provide a small illustrative example of block structure and rotation schedule. In this example, there are ten residents. For (PGY1, PGY2, PGY3, PGY4, PGY5), the number of blocks is (6, 7, 9, 8, 12) and each block contains (8 to 9, 7 to 8, 5 to 6, 6 to 7, 4 to 5) weeks respectively. Each resident is assigned to one department in each block. For example, resident A (a PGY5 resident) is assigned to Nights in Block 1 and HPB in Block 2. Each resident has a vacation week (highlighted in pink and marked with “v”).

#### 4. The Resident-to-Rotation Assignment Problem (RRAP)

In this section, we formally define the RRAP (Section 4.1). In Section 4.2, we present an IP formulation of the RRAP. Then, we present the equity-promoting counterpart in Section 4.3.

##### 4.1. Problem Setting

Consider a residency program that requires residents to complete different services during the academic year in different departments (hospital units). We define  $D$  as the set of departments,  $R$



as the set of residents, and  $E$  as the set of resident classes (e.g., PGY1, PGY2, ..., PGY5). Since each class may require a different training period in each department, the program usually divides the academic year of each class (typically 52–53 weeks at CUIMC) into several blocks. Each block consists of several consecutive weeks. The length of each block (i.e., the number of weeks) can be different. Moreover, some blocks may have more weeks than others. Assigning a resident to a *block* is equivalent to assigning this resident to a *rotation* period in a given department. For each class  $e \in E$ , we define  $B_e$  as the set of blocks for this class  $e$ ,  $R_e$  as the set of class  $e$  residents, and  $W_{b,e}$  as the set of weeks in each block  $b \in B_e$ . Residents of the same class have the same block arrangement, but the block arrangements could vary between classes. We define the following sets that could be customized depending on residency program requirements, ACGME regulations, and resident’s class and availability:

- $D_r^{req}$  is the set of *required departments* that resident  $r \in R$  must do, i.e., mandatory rotations.
- $D_r^{imp}$  is the set of *impossible departments* that residents  $r \in R$  cannot work in, i.e., resident  $r \in R$  cannot be assigned to any  $d \in D_r^{imp}$ .
- $D^{busy}$  is the set of *busy departments*. Such departments often have a heavy workload. Thus, if resident  $r \in R$  is assigned to serve in a department  $d \in D^{busy}$  in a week  $w \in W$ , then s/he cannot take a vacation during that week.
- $B_r^{imp}$  is the set of *impossible blocks* for resident  $r \in R$ , i.e., resident  $r \in R$  cannot be assigned to any block  $b \in B_r^{imp}$ .

Residents of the same and different classes could have different required and impossible departments. Moreover, program requirements and regulations often limit the number of weeks each resident can serve in each department. Accordingly, we define parameter  $T_{r,d}^{\min}$  ( $T_{r,d}^{\max}$ ) as the minimum (maximum) number of blocks each resident  $r \in R$  could work in department  $d \in D$ . On the other hand, each department may require a specific number of residents from each class. Accordingly, we define parameter  $R_{e,d,b}^{\min}$  ( $R_{e,d,b}^{\max}$ ) as the minimum (maximum) number of class  $e$ ’s residents required to serve in department  $d \in D$  in block  $b \in B_e$ .

Each resident should have a mandatory number of vacation weeks during the academic year (e.g., 2 or 4 weeks) that depends on ACGME rules and whether the resident is spending the entire academic year (e.g., categorical) or only a particular period (e.g., six months) in the program. Moreover, before the schedules are constructed, each resident submits their request for preferred vacation weeks. For each  $r \in R$ , we define  $T_r^{vac}$  as the number of mandatory vacation weeks and  $W_r^{vac}$  as the set of preferred vacation weeks. A complete list of our notation can be found in Table 2.

Given sets of residents, resident classes, departments, and blocks, as well as department types and staffing needs, rotation requirements for each class, and vacation requests, the RRAP consists of finding a feasible rotation schedule specifying each resident’s assignments to rotations and vacations. More intuitively, completing this scheduling problem can be visualized as filling the department name in each block as illustrated in Figure 1 and specifying vacation weeks for each resident.

Table 2: Notation (RRAP)

---

<b>Index sets</b>	
$E$	the set of residents' classes
$R_e$	the set of residents in class $e \in E$
$R$	the set of all residents, i.e., $R = \bigcup_{e \in E} R_e$
$B_e$	the set of blocks for residents in class $e$
$W_{b,e}$	the set of weeks in each block $b$ of residents in class $e$
$W$	the set of weeks in the planning horizon, i.e., $W = \bigcup_{b \in B_e} W_{b,e}, \forall e \in E$
$D$	the set of departments
$D_r^{\text{imp}}$	the set of resident $r$ 's impossible working department
$D_r^{\text{req}}$	the set of resident $r$ 's required working department
$D^{\text{busy}}$	the set of busy departments
$B_r^{\text{imp}}$	the set resident $r$ 's impossible working blocks
$W_r^{\text{vac}}$	the set of weeks that resident $r$ requests for vacations
<b>Parameters</b>	
$T_{r,d}^{\text{min}}$	resident $r$ 's minimum required working time (in blocks) in department $d$
$T_{r,d}^{\text{max}}$	resident $r$ 's maximum required working time (in blocks) in department $d$
$R_{e,d,b}^{\text{min}}$	minimum number of year $e$ 's residents required in department $d$ in block $b \in B_e$
$R_{e,d,b}^{\text{max}}$	maximum number of year $e$ 's residents required in department $d$ in block $b \in B_e$
$T_r^{\text{vac}}$	mandatory number of vacation weeks that resident $r$ should take
$D_{d,w}^{\text{vac}}$	maximum number of residents in vacation in department $d$ in week $W$ .
$R_{r,b}^{\text{vac}}$	maximum number of vacation weeks that a resident $r$ is allowed to take in block $b$ .
<b>Decision Variables</b>	
$z_{r,d,b}$	equals 1 if resident $r$ is assigned to department $d$ in block $b$
$x_{r,d,w}$	equals 1 if resident $r$ works in department $d$ in week $w$
$v_{r,d,w}$	equals 1 if resident $r$ has a vacation in week $w$ in department $d$

---

#### 4.2. The RRAP Formulation

In this section, we present our proposed IP formulation for the RRAP, which mimics the current manual method for generating the rotation schedule and provides the foundation for equity-promoting formulation presented in Section 4.3. We first introduce the sets, variables, and parameters defining this model. For each  $e \in E$ ,  $r \in R_e$ ,  $b \in B_e$ , and  $d \in D$ , we define a binary decision variable  $z_{r,d,b}$ , which equals 1 if resident  $r$  is assigned to department  $d$  in block  $b$ , and is zero otherwise. For each  $r \in R$ ,  $w \in W$ , and  $d \in D$ , we define a binary decision variable  $x_{r,d,w}$ , which equals 1 if resident  $r$  works in department  $d$  in week  $w$ , and is zero otherwise. Finally, we define a binary decision variable  $v_{r,d,w}$ , which equals 1 if resident  $r$  has a vacation in a week  $w$ , and is zero otherwise, for all  $r \in R$ ,  $w \in W$ , and  $d \in D$ . The RRAP formulation can be stated as the following feasibility problem.

$$\begin{aligned} & \text{maximize / minimize } 0 \\ & \mathbf{x, z, v} \end{aligned} \tag{1a}$$

$$\text{s.t. } \sum_{d \in D} z_{r,d,b} \leq 1, \quad \forall e \in E, r \in R_e, b \in B_e, \tag{1b}$$

$$\sum_{b \in B_e} z_{r,d,b} = 1, \quad \forall e \in E, r \in R_e, d \in D_r^{\text{req}}, \tag{1c}$$

$$z_{r,d,b} = 0, \quad \forall e \in E, r \in R_e, b \in B_e, d \in D_r^{\text{imp}}, \tag{1d}$$

$$z_{r,d,b} = 0, \quad \forall r \in R, b \in B_r^{\text{imp}}, d \in D, \tag{1e}$$

$$\sum_{w \in W} \sum_{d \in D} v_{r,d,w} = T_r^{\text{vac}}, \quad \forall r \in R, \quad (1f)$$

$$T_{r,d}^{\text{min}} \leq \sum_{b \in B_e} z_{r,d,b} \leq T_{r,d}^{\text{max}} \quad \forall e \in E, r \in R, d \in D_r^{\text{req}}, \quad (1g)$$

$$R_{e,d,b}^{\text{min}} \leq \sum_{r \in R_e} z_{r,d,b} \leq R_{e,d,b}^{\text{max}}, \quad \forall e \in E, d \in D, b \in B_e, \quad (1h)$$

$$v_{r,d,w} \leq 1 - z_{r,d,b}, \quad \forall e \in E, r \in R_e, b \in B_e, w \in W_{b,e}, d \in D^{\text{busy}}, \quad (1i)$$

$$\sum_{r \in R} v_{r,d,w} \leq D_{d,w}^{\text{vac}}, \quad \forall d \in D, w \in W, \quad (1j)$$

$$\sum_{w \in W_{b,e}} \sum_{d \in D} v_{r,d,w} \leq R_{r,b}^{\text{vac}}, \quad \forall e \in E, r \in R_e, b \in B_e, \quad (1k)$$

$$z_{r,d,b} = v_{r,d,w} + x_{r,d,w}, \quad \forall e \in E, r \in R_e, b \in B_e, d \in D, w \in W_{b,e}, \quad (1l)$$

$$z_{r,d,b}, x_{r,d,w}, v_{r,d,w} \in \{0, 1\}, \quad \forall e \in E, r \in R_e, d \in D, b \in B_e, w \in W_{b,e}. \quad (1m)$$

Formulation (1) finds a feasible rotation schedule that maximizes the total number of satisfied vacation requests. Constraints (1b) ensure that each resident is assigned to at most one department in each block. Constraints (1c) ensure that each resident is assigned to each required department exactly once. When a resident must serve in department  $d \in D$  more than one time, one can easily replace constraints (1c) by  $\sum_{b \in B_e} z_{r,d,b} = \text{Req}$ , where  $\text{Req}$  is the required number of rotations. Constraints (1d) ensure that residents are not assigned to rotations in impossible departments. Similarly, constraints (1e) ensure that residents are not assigned to impossible blocks. Constraints (1f) ensure that each resident has the required vacation weeks. Constraints (1g) ensure that the length of resident  $r$ 's rotation in department  $d$  satisfies the required minimum  $T_{r,d}^{\text{min}}$  and maximum  $T_{r,d}^{\text{max}}$  rotation length. Constraints (1h) ensure that the number of residents of class  $e$  working in department  $d$  satisfies  $R_{e,d,b}^{\text{min}}$  and  $R_{e,d,b}^{\text{max}}$  of that department. Recall that residents assigned to busy departments cannot take vacations during their rotation in these departments. Constraints (1i) ensure this condition. Constraints (1j) ensure that the total number of residents on vacation in department  $d$  and week  $w$  should be less than or equal to  $D_{d,w}^{\text{vac}}$ . Constraints (1k) ensure that each resident  $r \in R$  can take at most  $R_{r,b}^{\text{vac}}$  weeks of vacations in each block. Finally, constraints (1l) ensure that if resident  $r$  is assigned to department  $d$  in block  $b$ , s/he either works or has a vacation for each week within block  $b$ . We highlight that, due to constraints (1l), we can relax the binary restriction on  $\mathbf{v}$ .

We close this section by observing the following about formulation (1). First, we model rotation and vacation requirements as hard constraints that must be respected to adhere to CUIMC residency program requirements and ACGME rules. Relaxing these constraints may result in failure to meet rotation and educational standards and potentially cause the program to lose its accreditation. Second, formulation (1) provides a blind feasible assignment to rotations and vacations, potentially leading to disparities in satisfying vacation requests among residents, with some residents assigned

Table 3: Inequity measures

Index	Name of the measure	Mathematical expression
(1)	Range	$\max_{r \in R} u_r - \min_{r \in R} u_r$
(2)	Gini deviation (Gini)	$\sum_{r \in R} \sum_{r' \in R}  u_r - u_{r'} $
(3)	Maximum pairwise deviation (MaxPair)	$\max_{r \in R} \max_{r' \in R}  u_r - u_{r'} $
(4)	Absolute deviation from mean (MeanDev)	$\sum_{r \in R}  u_r - \bar{u} $
(5)	Maximum absolute deviation from mean (MaxMeanDev)	$\max_{r \in R}  u_r - \bar{u} $
(6)	Maximum sum of pairwise deviation (MaxSumPair)	$\max_{r \in R} \sum_{r' \in R}  u_r - u_{r'} $
(7)	Sum of Maximum pairwise deviation (SumMaxPair)	$\sum_{r \in R} \max_{r' \in R}  u_r - u_{r'} $

vacations in their preferred weeks and others assigned vacations in weeks different than their preferred weeks (see Section 7). Indeed, there is room to maximize the number of satisfied requests by considering the following formulation

$$\underset{\mathbf{v}, \mathbf{x}, \mathbf{z}}{\text{maximize}} \left\{ f_1(\mathbf{v}) := \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \mid (\mathbf{v}, \mathbf{x}, \mathbf{z}) \in \{(\mathbf{1b}) - (\mathbf{1m})\} \right\}. \quad (2)$$

However, as we later show in Section 7, similar to formulation (1), formulation (2) may result in a disparity in satisfying vacation requests because it has no measure to ensure fair vacation schedule. In what follows, we call formulation (2) an equity-neutral formulation. To mitigate this, in the next section, we derive an equity-promoting counterpart that maximizes the number of satisfied requests and minimizes a measure of inequity in the number of vacation requests granted based on residents' preferences. Finally, we note that residents of the same class will have the same workload in terms of the assignment to required rotations during the academic year; thus, there is no unfairness in this aspect.

#### 4.3. The Equity-Promoting Formulation

In this section, we derive an equity-promoting IP model for the RRAP. First, we introduce additional notations that are needed to define our equity-promoting model. We define  $u_r$  as the number of satisfied vacation requests for resident  $r \in R$ , i.e.,  $u_r = \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w}$ , where  $W_r^{\text{vac}}$  is the set of preferred vacation weeks for resident  $r$ . We let  $\bar{u} = \sum_{i=1}^{|R|} u_i / n$  represent the mean of  $\mathbf{u} := [u_1, \dots, u_{|R|}]^\top$ . Finally, we let  $\phi : \mathbb{R}^{|R| \times |D| \times |W|} \rightarrow \mathbb{R}$  represent an inequity measure. Table 3 presents well-known and commonly used inequity (unfairness) measures. We employ these measures to gauge the level of inequity among residents in terms of the number of satisfied vacation requests. For example, given a vacation schedule  $\mathbf{v}$ , we can compute the Gini deviation in the satisfied vacation requests as  $\phi(\mathbf{v}) = \sum_{r \in R} \sum_{r' \in R, r' \neq r} | \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} - \sum_{d \in D} \sum_{w \in W_{r'}^{\text{vac}}} v_{r',d,w} |$ . Note that  $\phi(\mathbf{v}) = 0$  implies perfect equity, whereas a larger value  $\phi(\mathbf{v}) > 0$  indicates a higher degree of inequity. Hence, to promote equity, one could consider minimizing  $\phi(\mathbf{v})$ , i.e.,

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{minimize}} \left\{ \phi(\mathbf{v}) \mid (\mathbf{z}, \mathbf{v}, \mathbf{x}) \in \{(\mathbf{1b}) - (\mathbf{1m})\} \right\}. \quad (3)$$

Formulation (3) finds an optimal rotation schedule that minimizes a measure of inequity in satisfied vacation requests. Our preliminary investigation suggests that such formulation leads to a rotation schedule with zero vacation requests satisfied. Intuitively, denying all vacation requests is the most equitable schedule, with  $\phi(\mathbf{v}) = 0$  for any  $\phi$ . While this is an equitable schedule, residents will be dissatisfied. Indeed, there is room to satisfy vacation requests and improve satisfaction while ensuring equity among residents. To do so, we incorporate both objectives: minimizing inequity in the number of satisfied vacation requests among residents and maximizing the number of satisfied requests. The resulting equity-promoting RRAP formulation is as follows.

$$\underset{\mathbf{v}, \mathbf{x}, \mathbf{z}}{\text{maximize}} \quad f_1(\mathbf{v}) := \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \quad (4a)$$

$$\underset{\mathbf{v}, \mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f_2(\mathbf{v}) := \phi(\mathbf{v}) \quad (4b)$$

$$\text{subject to:} \quad (\mathbf{v}, \mathbf{x}, \mathbf{z}) \in \{(\mathbf{1b}) - (\mathbf{1m})\}. \quad (4c)$$

Formulation (4) finds an optimal rotation schedule that simultaneously maximizes the total number of satisfied vacation requests (i.e.,  $f_1(\mathbf{v})$ ) and minimizes a measure of inequity in satisfied requests among residents (i.e.,  $f_2$ ). For brevity, we relegate the formulation of the form (4) based on each measure in Table 3 to [Online Appendix A](#). Note that one can employ formulation (4) to promote equity among residents irrespective of their classes or equity among residents of the same class (or group). In the former case, we evaluate inequity within the entire set of residents. In the latter case, we evaluate inequity within each group. For example, suppose that the program defines  $G$  groups of residents  $\{R_g\}_{g \in G}$  based on some criterion (e.g., seniority), where  $R_g$  is the set of residents belonging to group  $g$ . In this case, one can compute  $\phi_g(\mathbf{v})$  for each group of residents  $R_g$  and use  $f_2(\mathbf{v}) = \sum_{g \in G} \phi_g(\mathbf{v})$  in (4b). We compare these approaches in Section 7.4.

We observe the following about formulation (4). The two objectives  $f_1$  and  $f_2$  can be conflicting, i.e., improving one can entail deteriorating the other. Indeed, satisfying a larger number of vacation requests does not necessarily ensure equity in the number of satisfied vacation requests per resident. To see this, let us consider the following simple example. Suppose we have three residents (i.e.,  $R = \{1, 2, 3\}$ ), where each has one satisfied vacation request (i.e.,  $u_r = 1, \forall r \in R$  and  $f_1=3$ ). This is an equitable vacation assignment because all residents have the same number of satisfied requests. Indeed, the value of each inequity measure in Table 3 is zero under this vacation assignment, i.e.,  $f_2 = 0$ . Suppose we can satisfy an additional vacation request for one of the residents and we choose to fulfill one more request of resident 1. In this case,  $u_1 = 2, u_r = 1$  for  $r \in \{2, 3\}$ , and the total number of satisfied vacation requests increases to  $f_1 = 4$ . This is clearly inequitable as one resident has two requests granted while the other two have one request and the value of all inequity measures increases. For example, the Gini deviation (measure 2) increases from 0 (under the first assignment) to 4 (under the second assignment).

In the next section, we present methodologies for investigating the trade-off between  $f_1$  (number

of satisfied vacation requests) and  $f_2$  (inequity) and accordingly obtain Pareto-optimal rotation schedules. Here, Pareto optimal (equivalently, efficient, non-dominated, or non-inferior) rotation schedules or solutions to (4) are solutions that cannot be improved in one objective function without deteriorating their performance in the other one. The weighting and  $\varepsilon$ -constraint methods are widely used iterative methods to identify subsets of the Pareto optimal solutions for bi-objective IP problems, such as the equity-promoting RRAP. The  $\varepsilon$ -constraint method has several advantages over the weighting method. First, as pointed out by [Das and Dennis \(1997\)](#); [Ehrgott \(2005\)](#) and recently shown in [Bergman et al. \(2022\)](#), the weighting method often fails to find the complete set of Pareto optimal solutions to multi-objective integer and mixed integer programming problems. In contrast, the  $\varepsilon$ -constraint method does not suffer from this pitfall. Note that identifying the entire set of non-dominated rotation schedules is desirable since it maximizes the decision-maker’s knowledge about the trade-offs between the two objectives. They can then choose the “*most preferred*” schedule. Second, in the weighting method, the scaling of the objective functions strongly influences the obtained results. Therefore, one must scale the objective functions to a common scale before forming the weighted sum. In the  $\varepsilon$ -constraint method, this is not necessary. Given these shortcomings and our collaborator’s interest in analyzing the entire non-dominated rotation schedules, we do not adopt the weighting method.

## 5. Solution Methodology

In this section, we present our proposed method that produces the entire set of non-dominated rotation schedules of the equity-promoting RRAP problem in (4). In Section 5.1, we briefly discuss the traditional  $\varepsilon$ -constraint method and its challenges. Then, in Section 5.2, we present our proposed Pareto Search Algorithms. We relegate all proofs to [Online Appendix B](#).

### 5.1. $\varepsilon$ -Constraint Method

In this section, we briefly discuss the classical  $\varepsilon$ -constraint method (see [Ehrgott \(2005\)](#); [Balles-tero and Romero \(2013\)](#) for detailed discussions). To facilitate the discussion, we first introduce some notation and define relevant terms. We define the feasible set of problem (4) as  $\mathcal{F} = \{\mathbf{z} \in \mathbb{B}^{|R| \times |D| \times |B|}, \mathbf{x} \in \mathbb{B}^{|R| \times |D| \times |W|}, \mathbf{v} \in \mathbb{B}^{|R| \times |D| \times |W|} \mid (1b) - (1m)\}$ . Definitions 5.1 and 5.2 introduce the notions of Pareto optimal solutions and non-dominated points, respectively.

**Definition 5.1** (Pareto Optimal or Non-dominated Solutions). A feasible solution  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$  is called a Pareto-optimal (or non-dominated) solution if there is no other feasible solution  $(\mathbf{z}', \mathbf{x}', \mathbf{v}') \in \mathcal{F}$  such that  $f_1(\mathbf{v}') \geq f_1(\mathbf{v})$  and  $f_2(\mathbf{v}') \leq f_2(\mathbf{v})$ . We define the set of Pareto-optimal solutions as  $\mathcal{F}_p = \{(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F} \mid f_1(\mathbf{v}) \geq f_1(\mathbf{v}'), f_2(\mathbf{v}) \leq f_2(\mathbf{v}'), \forall (\mathbf{z}', \mathbf{x}', \mathbf{v}') \in \mathcal{F}\}$ .

**Definition 5.2** (Non-dominated Points). Given a Pareto-optimal solution  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}_p$ , the corresponding objective value  $(f_1(\mathbf{v}), f_2(\mathbf{v}))$  is called a non-dominate point. We call the set  $\mathcal{P} = \{(f_1(\mathbf{v}), f_2(\mathbf{v}))^\top \in \mathbb{R}^2 \mid (\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}_p\}$  the set of non-dominated points (Pareto front).

In the  $\varepsilon$ -constraint method, we optimize one objective function while constraining the value of

---

**Algorithm 1:** The  $\varepsilon$ -constraint method for the equity-promoting RRAP

---

**Initialization:**  $i = 1$ ,  $V = -1$ ,  $\varepsilon = 1$ ,  $\mathcal{P} = \emptyset$ ,  $\bar{V} =$  optimal value to (2)  
**while**  $V < \bar{V}$  **do**  
    • Solve problem (5). Record optimal solution  $(\mathbf{z}^*, \mathbf{x}^*, \mathbf{v}^*, f_1(\mathbf{v}^*))$  and value  $f_2(\mathbf{v}^*)$ .  
    • Update  $f_1^i \leftarrow f_1(\mathbf{v}^*)$ ,  $f_2^i \leftarrow f_2(\mathbf{v}^*)$ ,  $V \leftarrow f_1(\mathbf{v}^*)$ ,  $\mathcal{P} \leftarrow \mathcal{P} \cup \{(f_1^i, f_2^i)\}$ , and  $i \leftarrow i + 1$ .  
**end**

---

the other objective function, requiring that the objective value meets a threshold and iteratively adjusting that threshold. Specifically, the method produces the entire set of non-dominated solutions and points of problem (4) by solving a sequence of problems of the following form

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{minimize}} \{f_2(\mathbf{v}) \mid (\mathbf{z}, \mathbf{v}, \mathbf{x}) \in (\mathbf{1b}) - (\mathbf{1m}), f_1(\mathbf{v}) \geq V + \varepsilon\}, \quad (5)$$

where  $\varepsilon > 0$  is a pre-defined small positive constant. Algorithm 1 summarizes the steps of the  $\varepsilon$ -constraint method. To initialize the algorithm, we solve the equity-neutral problem (2) and record the optimal value  $\bar{V}$  (maximum number of vacation requests that can be satisfied). Then, the algorithm solves a sequence of problem (5) by successively adjusting (increasing) the parameter  $V$  in the RHS of the constrained objective  $f_1$ . Given that the minimum increase in the number of satisfied vacation requests in any feasible solution is one, we can set  $\varepsilon = 1$ . In each iteration, algorithm 1 seeks to identify a new feasible solution with one additional vacation request satisfied compared to the solution obtained from the previous iteration, where vacation requests are granted in a way that minimizes the inequity measure.

As is well-known, the traditional  $\varepsilon$ -constraint method may require substantial computational effort to generate the entire Pareto front (Kim and De Weck, 2005; Marler and Arora, 2010). Indeed, our results in Section 7.5 suggest that implementing the  $\varepsilon$ -constraint method to solve (4) with some inequity measures such as Gini and MeanDev is computationally challenging. In particular, as the number of fulfilled vacation requests, denoted as  $V$ , increases, problem 5 becomes extremely challenging to solve. To address this challenge, in the next section, we present our proposed Pareto Search algorithm, which, as shown in Section 7 can find the entire Pareto front for large RRAP instances within a reasonable time.

### 5.2. Pareto Search Algorithm

In this section, we present our proposed Pareto Search Algorithm. We first identify several characteristics of the optimal solutions and Pareto front, which we exploit in our proposed algorithm.

Recall that the minimum value of any inequity measure in Table 3 is zero. In Proposition 1, we show that there exists a Pareto-optimal solution  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}_p$  to (4) with  $f_2(\mathbf{v}^0) = 0$ .

**Proposition 1.** *An optimal solution  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0)$  to the optimization problem*

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{maximize}} \left\{ f_1(\mathbf{v}) \mid (\mathbf{z}, \mathbf{v}, \mathbf{x}) \in (\mathbf{1b}) - (\mathbf{1m}), f_2(\mathbf{v}) = 0 \right\} \quad (6)$$

*is a Pareto-optimal solution to problem (4), i.e.,  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}_p$  and  $(f_1(\mathbf{v}^0), f_2(\mathbf{v}^0)) \in \mathcal{P}$ .*

Recall that the optimal value to the equity-neutral formulation (2), denoted as  $\bar{V}$ , represents the maximum number of vacation requests that can be satisfied. In Proposition 2, we show there exists a Pareto-optimal solution  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}_p$  to (4) with  $f_1(\mathbf{v}^l) = \bar{V}$ .

**Proposition 2.** *Let  $\bar{V}$  be the optimal objective value of problem (2). An optimal solution  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l)$  to the optimization problem*

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{minimize}} \left\{ f_2(\mathbf{v}) \mid (\mathbf{z}, \mathbf{v}, \mathbf{x}) \in (1b) - (1m), f_1(\mathbf{v}) = \bar{V} \right\}, \quad (7)$$

*is a Pareto-optimal solution to problem (4), i.e.,  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}_p$  and  $(f_1(\mathbf{v}^l), f_2(\mathbf{v}^l)) \in \mathcal{P}$ .*

Propositions 1 and 2 indicate that any non-dominated point  $(f_1(\mathbf{v}), f_2(\mathbf{v})) \in \mathcal{P}$  associated with non-dominated solution  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}_p$  satisfies  $f_1(\mathbf{v}) \in [f_1(\mathbf{v}^0), f_1(\mathbf{v}^l)]$  and  $f_2(\mathbf{v}) \in [0, f_2(\mathbf{v}^l)]$ . Note also that the difference in the total number of satisfied requests between two non-dominated vacation schedules  $\mathbf{v} \in \mathcal{F}_p$  and  $\mathbf{v}' \in \mathcal{F}_p$  is at least one, i.e.,  $|f_1(\mathbf{v}) - f_1(\mathbf{v}')| \geq 1$ . Hence, given a non-dominated rotation schedule with a vacation schedule  $\mathbf{v}$  and  $V$  satisfied requests, the subsequent non-dominated rotation schedule might retain the same satisfied requests as in  $\mathbf{v}$  while fulfilling an additional request (totaling  $V + 1$  satisfied requests).

These observations motivate our Pareto Search Algorithm. Algorithm 2 summarizes the steps of this algorithm. First, we solve problem (6) and record optimal value  $f_1(\mathbf{v}^0)$  and solution  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0)$  with  $f_2(\mathbf{v}^0) = 0$ . We extract the following categories of residents from  $\mathbf{v}^0$ : the set of residents with one or more fulfilled requests denoted as  $\mathcal{V}^0$  and the set of residents with  $i$  satisfied vacation requests denoted as  $\mathcal{U}_i^0$ . For any vacation schedule  $\mathbf{v}^j$ , the sets  $\mathcal{V}^j$  and  $\mathcal{U}_i^j$  are defined as follows.

$$\mathcal{V}^j = \left\{ r \in R, d \in D, w \in W_r^{\text{vac}} : v_{r,d,w}^j = 1 \right\}. \quad (8)$$

$$\mathcal{U}_i^j = \left\{ r \in R : \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w}^j = i \right\}, \quad \forall i \in [0, \bar{w}]. \quad (9)$$

Parameter  $\bar{w}$  in (9) is the maximum number of requested weeks for vacation among all residents, i.e.,  $\bar{w} = \max_{r \in R} |W_r^{\text{vac}}|$ . In Step 2, we solve the equity-neutral model and record the optimal number of satisfied vacations  $f_1(\mathbf{v}^l)$ . Then, in each iteration  $j$  of Step 3 (with sets  $\mathcal{V}^0$  and  $\{\mathcal{U}_i^0\}_{i=0}^{\bar{w}}$  as inputs), the algorithm attempts to find a new feasible rotation schedule with a vacation schedule that has one more vacation request fulfilled than the vacation schedule obtained from the previous iteration. Specifically, for each  $V \in [f_1(\mathbf{v}^0) + 1, f_1(\mathbf{v}^l)]_{\mathbb{Z}}$ , Step 3 proceeds as follows. In Step 3.1, we identify the set of residents with the lowest number of satisfied vacation requests denoted as  $R'$ . In Step 3.2, we attempt to find an optimal schedule with one more satisfied vacation request than the previous iteration  $j - 1$  by solving the optimization problem

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{minimize}} \quad f_2(\mathbf{v}) \quad (10a)$$



---

**Algorithm 2:** The Pareto Search Algorithm
 

---

**Initialization:**  $j = 1$ ;  $\mathcal{P} = \emptyset$ ,  $\mathcal{F}_p = \emptyset$

**Step 1.** Solve problem (6) and record optimal solution  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0)$  and value  $(f_1(\mathbf{v}^0), f_2(\mathbf{v}^0))$ . Obtain sets  $\mathcal{V}^0$  and  $\{\mathcal{U}_i^0\}_{i=0}^{\bar{w}}$  using (8) and (9). Set  $\mathcal{F}_p = \{(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0)\}$ ,  $\mathcal{P} = \{(f_1(\mathbf{v}^0), 0)\}$ , and  $V = f_1(\mathbf{v}^0) + 1$

**Step 2.** Solve the equity-neutral RRAP problem (2) and record optimal solution  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l)$  and value  $(f_1(\mathbf{v}^l), f_2(\mathbf{v}^l))$ .

**Step 3. Obtain Optimal Solutions**

**while**  $V < f_1(\mathbf{v}^l)$  **do**

3.1 Set  $s \leftarrow \arg \min_{i \in [0, \bar{w}]} \{U_i^{j-1} \neq \emptyset\}$  and  $R' \leftarrow U_s^{j-1}$

3.2 Solve problem (10)

**if** problem (10) has an optimal solution  $(\mathbf{z}^*, \mathbf{x}^*, \mathbf{v}^*, f_1(\mathbf{v}^*))$  and value  $f_2(\mathbf{v}^*)$  **then**

Update sets  $\mathcal{V}^j$  and  $\{\mathcal{U}^j\}_{i=0}^{\bar{w}}$ . Set  $f_1^j \leftarrow f_1(\mathbf{v}^*)$ ,  $f_2^j \leftarrow f_2(\mathbf{v}^*)$ ,  $S^j \leftarrow \{(\mathbf{z}^*, \mathbf{x}^*, \mathbf{v}^*)\}$ ;

$V \leftarrow V + 1$  and  $j \leftarrow j + 1$ . **Go to Step 3.1** ;

**else Go to Step 3.3;**

3.3 Solve problem (11)

**if** problem (11) has an optimal solution  $(\mathbf{z}^*, \mathbf{x}^*, \mathbf{v}^*, f_1(\mathbf{v}^*))$  and value  $f_2(\mathbf{v}^*)$  **then**

Update sets  $\mathcal{V}^j$  and  $\{\mathcal{U}^j\}_{i=0}^{\bar{w}}$ . Set  $f_1^j \leftarrow f_1(\mathbf{v}^*)$ ,  $f_2^j \leftarrow f_2(\mathbf{v}^*)$ ,  $S^j \leftarrow \{(\mathbf{z}^*, \mathbf{x}^*, \mathbf{v}^*)\}$ ;

$V \leftarrow V + 1$  and  $j \leftarrow j + 1$ . **Go to Step 3.1** ;

**else** Set  $V \leftarrow V + 1$  and **go to Step 3.3.**;

**end**

**Step 4. Extract Non-dominated Solutions**

**for**  $k \in [1, j]$  **do**

Let  $I = \{i \in [1, j] | f_1^i \geq f_1^k\}$ ;

**if**  $f_2^k < f_2^i \quad \forall i \in I$  **then**

|  $\mathcal{P} \leftarrow \mathcal{P} \cup \{(f_1^k, f_2^k)\}$ ,  $\mathcal{F}_p \leftarrow \mathcal{F}_p \cup S^k$

**end**

**end**

---

$$\text{subject to: } (1b) - (1m), \tag{10b}$$

$$f_1(\mathbf{v}) = V, \tag{10c}$$

$$v_{r,d,w} = 1, \quad \forall (r, d, w) \in \mathcal{V}^{j-1}, \tag{10d}$$

$$\sum_{r \in R'} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} = 1. \tag{10e}$$

Problem (10) aims to find an optimal rotation schedule with  $f_1 = V$  satisfied vacation requests that minimize a measure of inequity in the satisfied vacation requests ( $f_2 = \phi(\mathbf{v})$ ). The associated vacation schedule has the same satisfied vacation requests as the one obtained from  $(j-1)$ , with one additional vacation request being satisfied. This feature, which is desirable since, for the purpose of comparison, it is preferred to have solutions throughout the Pareto front that involve common assignments rather than very distinct ones, is ensured via constraints (10c)–(10e). Specifically, constraints (10c) ensure the number of satisfied vacation requests is  $V$ . Constraints (10d) ensure that satisfied vacation requests for residents  $r \in \mathcal{V}^{j-1}$  in the schedule obtained in iteration  $(j-1)$

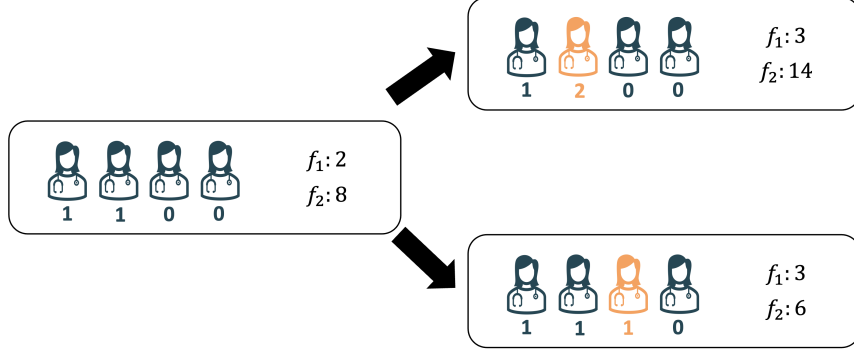


Figure 2: An illustration of Proposition 3 using an instance of four residents and the Gini deviation as the inequity measure  $f_2$ . The left panel shows a vacation schedule in which two residents have one fulfilled request while the other two have no fulfilled requests, so  $f_1 = 2$  and  $f_2 = 8$  in this schedule. Suppose we can satisfy one more request. The right panel shows two options. The first option is to fulfill a vacation request of one of the two residents who already have one satisfied request. In this case,  $f_1 = 3$  and  $f_2 = 14$ . The second option is to fulfill a vacation request for one of the residents with zero satisfied requests (i.e., with the fewest satisfied requests). In this case,  $f_1 = 3$  and  $f_2 = 6$ . Thus, the second option results in the lowest value of the inequity measure  $f_2$ .

remain satisfied in the new schedule. Constraints (10e) guarantee that we prioritize satisfying an additional vacation request of one of the residents  $r \in R'$  with the fewest granted vacation requests in the previous schedule  $j - 1$  (identified in Step 3.1). Proposition 3 establishes that fulfilling a vacation request for one of the residents with the fewest satisfied requests is always optimal; see Online Appendix B.3 for a proof and Figure 2 for an illustrative example.

If problem (10) has an optimal solution, we store the optimal solution and value, update sets  $\mathcal{V}^j$  and  $\mathcal{U}^j$  using (8) and (9), set  $V \leftarrow V + 1$  and  $j \leftarrow j + 1$ , and return to Step 3.1. If problem (10) does not have a feasible solution, this indicates that we cannot satisfy an additional vacation request based on the current vacation schedule  $\{v_{r,d,w} = 1, \forall (r,d,w) \in \mathcal{V}^{j-1}\}$ . In this case, we proceed to Step 3.3, where, we attempt to solve the following problem

$$\underset{\mathbf{z}, \mathbf{v}, \mathbf{x}}{\text{minimize}} \left\{ f_2(\mathbf{v}) \mid (\mathbf{z}, \mathbf{v}, \mathbf{x}) \in (1b) - (1m), f_1(\mathbf{v}) = V \right\}. \quad (11)$$

If problem (11) has an optimal solution, we record the optimal solution and value, update sets  $\mathcal{V}^j$  and  $\mathcal{U}^j$ , set  $V \leftarrow V + 1$  and  $j \leftarrow j + 1$ , and return to Step 3.1. On the other hand, if problem (11) does not have a feasible solution, this indicates that there is no optimal rotation schedule with exactly  $V$  satisfied vacation requests. In this case, we increase  $V$  by 1 and try to solve problem (11) again. Step 3 terminates when  $V = f_1(\mathbf{v}_l)$ . Note that Step 3 generates all non-dominated solutions and some dominated solutions. In Step 4, we extract non-dominated solutions. Given that  $V$  is finite, then Step 3 and hence Algorithm 2 terminates in a finite number of iterations.

**Proposition 3.** *Consider a vacation schedule  $\mathbf{v}$  and let  $R'$  be the set of residents with the fewest number of approved requests in  $\mathbf{v}$ . Suppose we can fulfill one additional vacation request. Then, fulfilling a vacation request for one of the residents  $r \in R'$  yields the lowest value of any inequity measure in Table 3.*

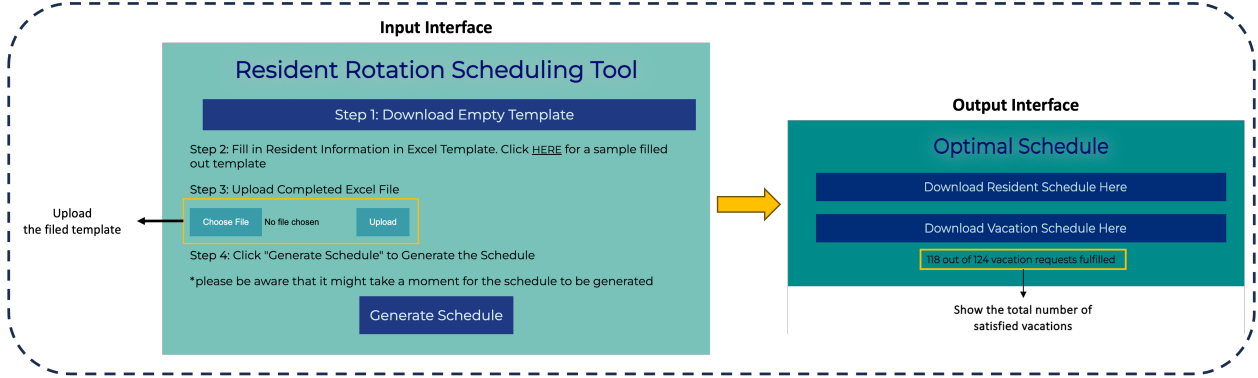


Figure 3: User Interface of the RRAP Tool

Solving problem (10) in Step 3.2 of Algorithm 2 is much easier than solving problem (11) in Step 3.3. Both problems aim to find a new rotation schedule with one more satisfied vacation request than the one found in the previous iteration while minimizing the value of the inequity measure, i.e.,  $f_2 = \phi$ . However, the search space in (10) is smaller because we fix  $\{v_{r,d,w} = 1, \forall (r, d, w) \in \mathcal{V}^{j-1}\}$  via constraint (10d) and approve a request of one of the residents with the fewest approved requests via constraint (10e). It is easy to verify that an optimal solution to (10) is also optimal to (11). Similarly, solving problem (10) is also much easier than solving problem (5) in each iteration of Algorithm 1.

## 6. The RRAP Tool

We developed a web-based tool implementing the proposed models using the Python programming language to automate the generation of the annual rotation schedule. Figure 3 presents a screenshot of the user interface, which has two modules: the input interface and the output interface. Details of these components are provided next.

The input interface is used to input data (i.e., details of an RRAP instance) required for solving the IP model. Specifically, the user first downloads the info template (Step 1), an Excel workbook that has several sheets, each designated for a specific set of input parameters to the IP model (e.g., program information, block information, list of residents, hospital departments, mandatory and impossible departments, etc.); see [Online Appendix C](#) for an example. Once the user fills out the info template, they upload it using the upload function (Step 3). Then, they click on the “Generate Schedule” function to generate the rotation schedule. Specifically, in the back end, this function reads the data from the filled input template, uses it as input to the IP model, and then calls Gurobi to solve the model. The output interface enables users to download the optimal solution using two Excel workbooks. The first provides the rotation schedule, and the second provides the associated vacation schedule.

## 7. Numerical Results

In this section, we construct various RRAP instances and conduct extensive experiments to illustrate the computational efficiency of the proposed model; compare optimal solutions to the

Table 4: Number of blocks and possible departments for each class in each RRAP Instances

Inst	Blocks					Departments				
	PGY1	PGY2	PGY3	PGY4	PGY5	PGY1	PGY2	PGY3	PGY4	PGY5
Inst1	12	8	9	7	6	12	6	8	7	6
Inst2	12	12	12	12	12	12	12	12	12	12
Inst3	12	12	10	10	9	12	10	10	10	9

proposed models and their impact on equity; investigate the trade-off between equity in satisfying vacation requests and the number of satisfied requests; and derive insights relevant to practice. In Section 7.1, we discuss experimental setups. In Section 7.2, we analyze the trade-off between inequity and the number of satisfied requests under different inequity measures. In Section 7.3, we compare our proposed approach with the manual approach. In Section 7.4, we compare the residents-based and class-based approaches. Finally, in Section 7.5, we analyze the computational performance of our approaches and the  $\varepsilon$ -constraint method.

### 7.1. Test Instances and Experimental Setup

To show our proposed approach’s broad applicability and benefits, we constructed various RRAP instances based on the data provided by CUIMC’s general surgery residency program. As detailed in Section 3, this program is a five-year-long program, where each academic year consists of  $|W| = 53$  weeks. We constructed three sets of RRAP instances based on the data from six academic years (2018–2023) by varying the number of blocks and departments for each resident class.

Table 4 summarizes the number of blocks and departments for each class in each instance. The number of weeks in each block in each instance is as follows. For **Inst1**, the number of weeks in each block of (PGY1, PGY2, PGY3, PGY4, PGY5) is (4 to 5, 6 to 7, 6 to 7, 7 to 9, 9 to 10). For **Inst2**, the number of weeks in each block of PGY1–PGY5 is 4 to 5 weeks. For **Inst3**, the number of weeks in each block of (PGY1, PGY2, PGY3, PGY4, PGY5) is (4 to 5, 4 to 5, 5 to 6, 5 to 6, 5 to 6). Table 5 summarizes the sets of mandatory/required (i.e.,  $D^{\text{req}}$ ) and busy departments (i.e.,  $D^{\text{busy}}$ ), where busy departments are highlighted in bold text. The set of impossible departments includes those that are not mandatory, i.e.,  $D_r^{\text{imp}} = D \setminus D_r^{\text{req}}, \forall r \in R$ . Note that **Inst2** represents a residency program where all resident classes have the same sets of blocks and mandatory departments. In contrast, **Inst1** and **Inst3** represent programs where resident classes have different rotation requirements. We provide details of departmental staffing needs ( $R^{\text{min}}$  and  $R^{\text{max}}$ ) and the required rotation length ( $T^{\text{min}}$  and  $T^{\text{max}}$ ) in [Online Appendix D](#).

We generate vacation requests for each resident using two methods, denoted as A and B. In method A, we randomly select `vacreq` non-overlapping weeks from the 53 weeks for each resident. In method B, we first partition the academic year evenly into three parts, denoted as  $W_1$ ,  $W_2$ , and  $W_3$ . Then, we randomly generate vacation requests as follows: 20% of residents’ vacation requests are drawn from  $W_1$ , 30% of requests are drawn from  $W_2$ , and the remaining 50% of vacation requests are from  $W_3$ . Based on these methods and **Inst1–Inst3**, we construct 18 RRAP instances with  $|R| \in \{62, 125, 185\}$  residents, where instances with  $|R| = 125$  and 185 represent large instances as

Table 5: Mandatory and busy departments for each class in each RRAP Instance. Busy departments are highlighted in bold text.

Inst	Class	Departments
Inst1	PGY1	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
	PGY2	Overlook, ACS-OR, Consults, CTICU, Peds, Allen
	PGY3	Renal, Lap, CR, Trauma, Overlook, Vascular, Breast, Thoracic
	PGY4	Overlook, Elective, Vascular, HPB, ACS, Allen, <b>Nights</b>
	PGY5	HPB-Chabot, CR, HPB, Lap, Elective, <b>Nights</b>
Inst2	PGY1	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
	PGY2	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
	PGY3	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
	PGY4	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
	PGY5	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>Rainbow</b> , <b>VTF</b>
Inst3	PGY1	Allen, Vascular, Breast, Thoracic, CR, SICU, HPB, Peds, Overlook, Lap, <b>VTF</b> , <b>Rainbow</b>
	PGY2	Allen, Vascular, Breast, VTF, Thoracic, CR, HPB, Consults, <b>ACS</b> , <b>Overlook</b>
	PGY3	Allen, Vascular, Breast, VTF, Thoracic, CR, SICU, HPB, Trauma, <b>Overlook</b>
	PGY4	Allen, Vascular, Breast, Lap, Thoracic, CR, SICU, HPB, <b>Overlook</b> , <b>Nights</b>
	PGY5	Allen, Vascular, Breast, Lap, Thoracic, CR, SICU, ACS, <b>Nights</b>

Allen: Allen Hospital; CR: Colorectal Surgery; SICU: Surgical Intensive Care Unit; HPB: Hepatopancreaticobiliary; Peds: Pediatric Surgery; Rainbow: Night Float for PGY1; Lap: Advanced Laparoscopic Surgery; Overlook: Overlook Hospital; ACS-OR: Acute Care Surgery Operative; Consults: Acute Care Surgery Consult; CTICU: Cardiothoracic Intensive Care Unit; Renal: Renal Transplant; Nights: Night Float; VTF: Veterinary Treatment Facility.

the typical number of general surgery residents at CUMIC often ranges from 50–65. We denote each instance as **Inst#-R-Method**. For example, **Inst1-62-A** is **Inst1** with 62 residents and vacation requests generated using method **A**.

Residents are required to stay in the program for the entire academic year (i.e.,  $B_r^{\text{imp}} = \emptyset, \forall r \in R$ ). Moreover, ACGME requires each resident to have four vacation weeks during the academic year. Accordingly, we set  $|W_r^{\text{vac}}| = 4$  in all instances. Each resident could submit two vacation requests at the beginning of the year (i.e.,  $T_r^{\text{vac}} = 2, \forall r \in R$ ), and each can have at most one vacation in each block/rotation (i.e.,  $R_{r,b}^{\text{vac}} = 1, \forall r \in R, b \in B$ ). Moreover, for instances with 62 and 125 residents, at most, one resident can be on vacation per week in each department, i.e.,  $D_{d,w}^{\text{vac}} = 1$ , for all  $d \in D$  and  $w \in W$ . For instances with 185 residents, at most three residents can be on vacation per week in each department, i.e.,  $D_{d,w}^{\text{vac}} = 3$ , for all  $d \in D$  and  $w \in W$ .

We use the constructed 18 RRAP instances (**Inst1-62-A**, **Inst1-62-B**, ..., **Inst3-185-B**) to analyze the computational time of the proposed methodologies in Section 7.5. In addition, we use a case study based on the data related to the 2023–2024 academic year to derive the practical insights discussed in Sections 7.2–7.4. We call this instance **RRAP-Case**. Specifically, this instance consists of 54 residents, of which (22, 9, 9, 7, 7) are (PGY1, PGY2, PGY3, PGY4, and PGY5) residents. We consider five types of PGY1 residents in this instance<sup>1</sup>, namely PGY1-Categorical, PGY1-OMFS, PGY1-Prelim, PGY1-GU, and PGY1-Ortho. PGY1-Categorical and PGY1-Prelim

<sup>1</sup>Note that there are typically eight types of PGY1 residents: Categorical, Prelim, GU, OMFS, Ortho, Cardiac, ENT, and Plastics. We do not include Cardiac, ENT, and Plastics in this illustrative instance.

are general surgery residents with categorical and preliminary positions, respectively (see Section 3 for details). On the other hand, PGY1-GU, PGY1-OMFS, and PGY1-Ortho are from the urology, oral and maxillofacial, and orthopedic surgical residency programs, respectively, rotating within the general surgery program. PGY2 and PGY3 residents are either Categorical or Cardiac. All residents, except PGY1-GU and PGY1-Ortho, rotate in the program for the entire academic year, while PGY1-GU and PGY1-Ortho spend 40 and 26 weeks, respectively. Moreover, each resident, except PGY1-Ortho, have four vacation weeks ( $T_r^{\text{vac}} = 4$ ), while PGY1-Ortho (because they are only present half of the year) have two vacation weeks ( $T_r^{\text{vac}} = 2$ ). Each resident can submit requests for  $|W_r^{\text{vac}}| = T_r^{\text{vac}}$  preferred vacation weeks. The sets of blocks (mandatory, possible, and busy) departments for each class and each type of resident are different. We refer to [Online Appendix G](#) for details and parameter settings of this instance.

In our implementation, we add the following symmetry-breaking constraints (12) to ensure that, for residents staying in the program for the whole academic year (i.e.,  $\{r \in R \mid B_r^{\text{imp}} = \emptyset\}$ ), their block schedules are filled sequentially (i.e., a department should be assigned to block  $b$  before block  $b + 1$ ):

$$\sum_{d \in D} z_{r,d,b} \geq \sum_{d \in D} z_{r,d,b+1}, \quad \forall r \in R : B_r^{\text{imp}} = \emptyset, b \in B. \quad (12)$$

We implemented our proposed models and algorithm in Python 3.8.6 and used Gurobi 9.5.0 as the solver with default settings. We conducted all the experiments on a MacBook Pro with an M1 chip and 32 gigabytes (GB) RAM. This represents the computing capability available to a chief resident who would be tasked with producing a schedule in a real-world setting.

### 7.2. Analysis of the Trade-off between Inequity and Number of Satisfied Requests

In this section, we analyze the trade-off between inequity ( $f_2$ ) and the number of satisfied vacation requests ( $f_1$ ) using the **RRAP-Case** instance. Figure 4 presents the Pareto fronts (trade-off curves) obtained by solving this instance using the Pareto Search Algorithm with different inequity measures. The  $(x, y)$  values of each point on the curve are the value of  $(f_1, f_2)$ =(total number of satisfied requests, value of the inequity measure) associated with a Pareto optimal rotation schedule. The red star represents the values of  $(f_1, f_2)$  associated with a feasible rotation schedule obtained by solving the feasibility problem (1).

Figure 4 provides several important insights. First, all non-dominated points lies between  $[108, 0]$  and  $[193, f_2]$ , where 108 is the optimal value of problem (6) and  $f_2$  is the optimal value of problem (4), and  $f_1=193$  is the optimal value of the equity-neutral problem (2). This is consistent with our theoretical results in Propositions 1–2. Second, the equity-neutral model produces rotation schedules that maximize both the number of satisfied requests and the value of each inequity measure. This shows that these schedules exhibit significant disparities in meeting residents’ vacation requests.

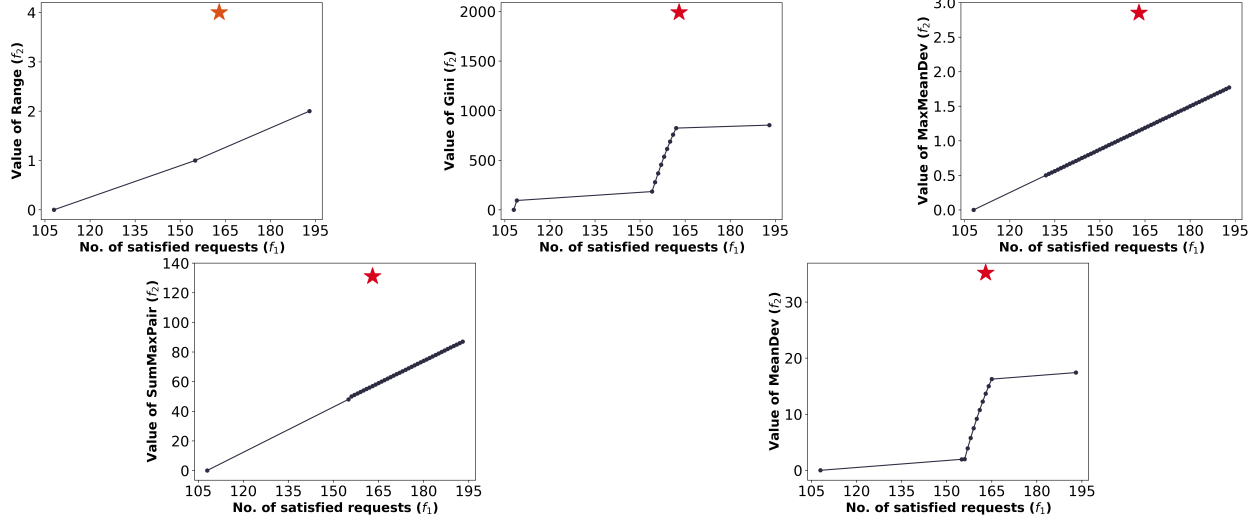


Figure 4: Pareto fronts illustrating the trade-off between the number of satisfied vacation requests ( $f_1$ ) and inequity ( $f_2$ ). The red star represents a feasible rotation schedule.

Third, it is clear that different choices of the inequity measure in the equity-promoting model produce different sets of Pareto optimal (non-dominated) rotation schedules with varying impacts on equity and the number of satisfied requests. Moreover, the number of non-dominated rotation schedules and points varies under different measures. Specifically, the number of non-dominated points under (Range, Gini, MeanDev, MaxMeanDev, SumMaxPair) = (3, 12, 13, 63, 40). Note also that the non-dominated points are different. For example, using the Gini deviation, we can find a rotation schedule with 109 satisfied requests and a Gini deviation of 94, i.e.,  $(f_1, f_2) = (109, 94)$ . But there is no such rotation schedule under other measures. Although there is no clear winner among these inequity measures or criteria for selecting any of them, the program director and chief resident of the collaborating health system favor utilizing measures that identify a larger number of non-dominated solutions. This choice allows for more rotation scheduling options and flexibility in selecting a preferred schedule while acknowledging the trade-offs between the two objectives.

Finally, we observe that the optimal equity-promoting rotation schedules always dominate the one obtained by solving feasibility problem (1) (represented by a star in Figure 4). In particular, using a feasible solution, we can satisfy 163 vacation requests, and the corresponding values of (Range, Gini, MeanDev, MaxMeanDev, and SumMaxPair) are (4, 1990, 35.2, 2.9, 131). These points are notably above the Pareto fronts. Moreover, we can find non-dominated rotation schedules with more satisfied requests and lower values of each measure.

### 7.3. Comparison with the Sequential Approach

In practice, the chief resident or program director employs a sequential approach, separating rotation assignment decisions ( $\mathbf{z}$ ) from the remaining decisions ( $\mathbf{v}$  and  $\mathbf{x}$ ). Specifically, first, they assign residents to blocks and departments (i.e., find a feasible  $\mathbf{z}$ ). Then, they construct a vacation schedule, specifying vacation and working weeks for each resident (i.e., find feasible

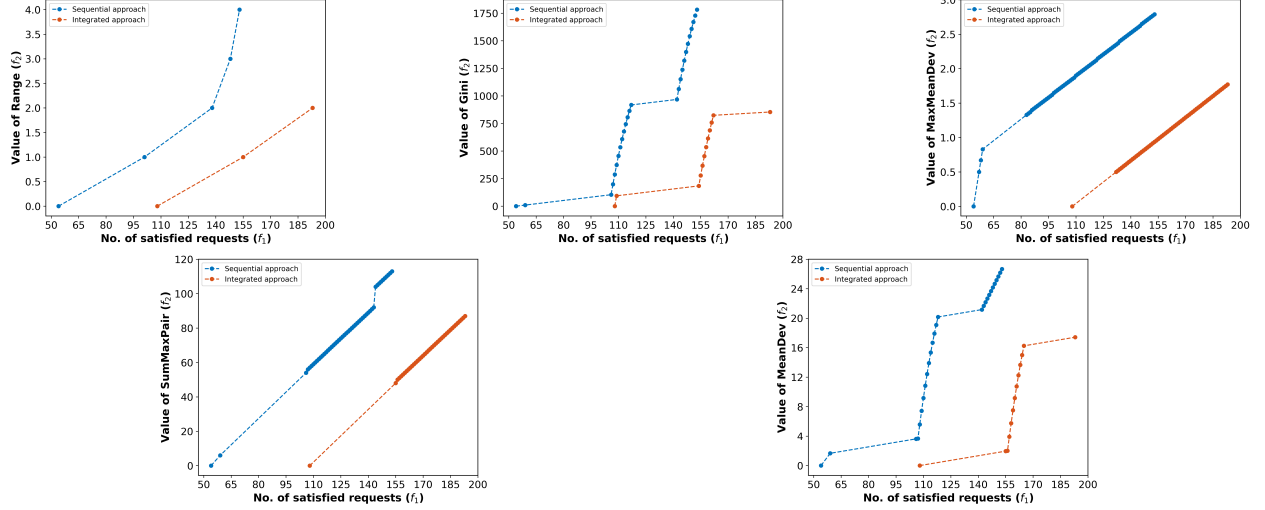


Figure 5: Pareto fronts resulting from the integrated and sequential approaches.

$\mathbf{v}$  and  $\mathbf{x}$ ). In this section, we compare the performance of this sequential approach with our proposed integrated approach. In the sequential approach, we first solve a feasibility problem  $\max_{\mathbf{z}} \{0 | (\mathbf{1b}) - (\mathbf{1e}), (\mathbf{1g}) - (\mathbf{1h}), \mathbf{z} \in \{0, 1\}^{|R| \times |D| \times |B|}\}$  to obtain a feasible  $\bar{\mathbf{z}}$ . Then, we solve the following equity-promoting IP with  $\mathbf{z}$  fixed to  $\bar{\mathbf{z}}$  to obtain corresponding  $(\mathbf{v}, \mathbf{x})$  and  $(f_1, f_2)$ .

$$\underset{\mathbf{v}, \mathbf{x}}{\text{maximize}} \quad f_1(\mathbf{v}) := \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \quad (13a)$$

$$\underset{\mathbf{v}, \mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f_2(\mathbf{v}) := \phi(\mathbf{v}) \quad (13b)$$

$$\text{subject to:} \quad (\mathbf{v}, \mathbf{x}) \in \{(\mathbf{1f}), (\mathbf{1i}) - (\mathbf{1m})\} \quad (13c)$$

Figure 5 presents the Pareto fronts obtained by solving the RRAP-Case instance using our integrated approach and the sequential approach. We observe that the sequential approach results in inferior solutions, which are dominated by those obtained using our proposed approach. In particular, our approach allows us to find equitable rotation schedules with a larger number of satisfied vacation requests and a lower value of the inequity measure. These results demonstrate the importance of integrating rotation and vacation scheduling decisions and show how our integrated equity-promoting approach can yield more equitable rotation schedules than the equity-promoting sequential approach, which separates rotation and vacation scheduling decisions.

#### 7.4. Residents-based Versus Class-based Equity

Recall from Section 4.3 that one can employ the equity-promoting model (4) to promote equity among all residents or within residents of the same class. In the former approach (denoted as the residents-based), we evaluate inequity across the entire set of residents, and in the latter (denoted as the class-based), we evaluate inequity within each class of residents. In this section, we compare solutions obtained from these approaches.

For brevity and illustrative purposes, in Figure 6, we present Pareto fronts obtained using



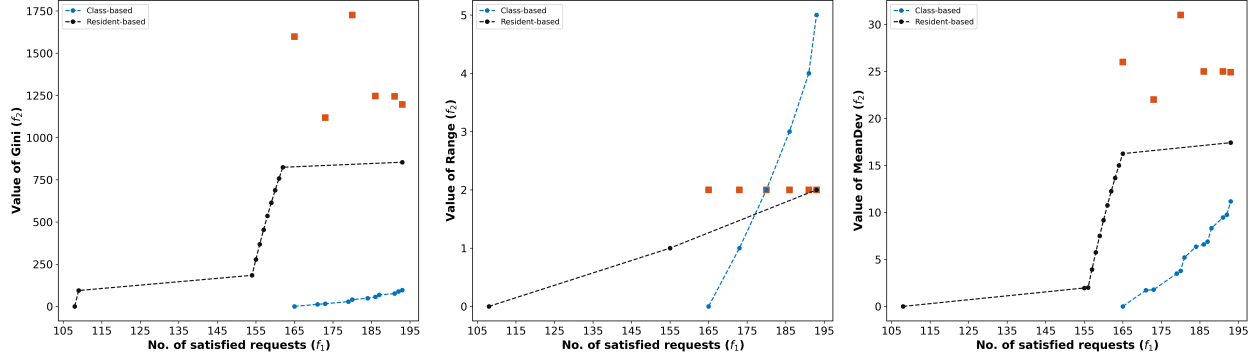


Figure 6: Pareto fronts resulting from residents-based and class-based approaches.

the residents-based (black curve) and class-based (blue curve) approaches with Range, Gini, and MeanDev. The  $f_2$  value of each point on the class-based curve is computed as  $f_2(\mathbf{v}) = \sum_{c \in C} \phi_c(\mathbf{v})$ , where  $C$  is the set of resident classes, i.e.,  $C = \{\text{PGY1}, \text{PGY2}, \text{PGY3}, \text{PGY4}, \text{PGY5}\}$ , and  $\phi_c(\mathbf{v})$  the value of the inequity measure for each class. We also evaluate the impact of employing class-based schedules on inequity among all residents by computing the value of the inequity measure considering the entire set of residents as in the residents-based approach. Red squares in Figure 6 represent the resulting  $(f_1, f_2)$  values.

It is clear that the residents- and the class-based approaches result in distinct sets of rotation schedules. This makes sense because they consider different sets of residents when measuring inequity. Moreover, while the class-based approach improves equity within each class, it potentially results in significant disparities among residents of different classes compared with the residents-based approach. In particular, rotation schedules generated using the class-based approach result in high values of each inequity measure when computed considering all residents. These values are significantly higher than those associated with rotation schedules obtained using the residents-based approach (see red squares in Figure 6).

It is worth noting that there is no universal agreement in the literature regarding the preference for either an individual-based approach (as in the residents-based) or a group-based approach (as in the class-based). The choice often depends on the context and is a matter of subjective evaluation. The results in this section indicate that, in terms of promoting equity among all residents, regardless of their class, the class-based approach may not be suitable.

### 7.5. Computational Performance

In this section, we analyze the solution time of the proposed approaches. We first generate five instances of each of the 18 RRAP instances (*Inst1-62-A*, *Inst1-62-B*, ..., *Inst3-185-B*) described in Section 7.1 for a total of 90 instances and then solve each with the proposed models and the Pareto Search Algorithm.

Let us first analyze the computational performance of the Pareto Search Algorithm (Algorithm 2). Table 6 presents the total solution time required to generate the complete set of Pareto

Table 6: The total time required to obtain the set of non-dominated solutions using Algorithm 2.

Instance	Range	Gini	MeanDev	MaxMeanDev	SumMaxPair
Inst1-62-A	140.1	239.5	235.4	174.9	218.1
Inst1-62-B	208.0	239.5	527.0	241.0	546.8
Inst2-62-A	118.6	301.2	157.2	95.5	200.2
Inst2-62-B	143.9	222.8	311.1	158.7	500.8
Inst3-62-A	110.0	239.5	183.9	139.5	177.1
Inst3-62-B	154.3	644.1	333.6	182.4	434.6
Inst1-125-A	557.8	1179.2	1188.7	853.1	1267.0
Inst1-125-B	920.1	1222.4	1265.0	1047.1	3778.2
Inst2-125-A	699.3	2874.3	1703.2	1360.6	1016.5
Inst2-125-B	1250.0	3528.7	2028.7	2586.9	1370.3
Inst3-125-A	496.4	1243.5	1020.4	877.0	953.0
Inst3-125-B	704.3	1834.5	1150.8	942.4	1589.7
Inst1-185-A	1209.5	4077.0	3344.0	2091.8	1267.0
Inst1-185-B	1637.6	4491.1	4342.1	2475.4	3825.4
Inst2-185-A	1350.3	6240.8	3252.8	7273.5	2324.0
Inst2-185-B	1772.1	5971.6	3737.1	2797.4	4005.5
Inst3-185-A	917.5	4091.1	2597.9	1882.4	2184.0
Inst3-185-B	2932.2	7972.1	7807.7	3315.1	6190.3

optimal rotation schedules to the equity-promoting model with measures Range, Gini, MeanDev, MaxMeanDev, and SumMaxPair. We do not present solution times with measures MaxPair and MaxSumPair since, as shown in the Proof of Proposition 3 in [Online Appendix B](#), the former measure is equivalent to Range, and the latter is equivalent to MaxMeanDev and thus have comparable solution times. In [Online Appendix E](#), we present the average time spent in each step of Algorithm 2.

We observe the following from Table 6. First, we can obtain the set of Pareto optimal solutions to the equity-promoting model using Algorithm 2 in less than two hours, irrespective of the inequity measure used in the model. Second, the algorithm takes a longer time as the instance size increases. Specifically, the ranges of the total time required to obtain the complete set for instances with 62, 125, and 185 residents are approximately 1.6–9 minutes, 12–62 minutes, and 15 minutes–2.2 hours. These results make sense because the size of the equity-promoting formulation and thus problems (10) and (11) in Step 3 of the algorithm increase with  $|R|$ , potentially increasing the time required to solve each in each iteration of the algorithm. Moreover, instances with more residents have more vacation requests and, hence, have larger numbers of non-dominated solutions, potentially increasing the time required to identify these solutions. However, these solution times are suitable for practical implementation, considering that the rotation schedule is constructed once a year and CUIMC typically has a range of 50 to 65 general surgery residents.

Third, the algorithm’s computational performance varies depending on the inequity measure employed in the equity-promoting model. In particular, the algorithm takes a shorter time to find the non-dominated set under the Range than under the other considered measures, and it takes the longest time under the Gini deviation. Intuitively, different measures require introducing different sets of additional variables and constraints into the model; see [Online Appendix A](#). Thus, the size

and complexity of solving the resulting formulation under each measure differ. For example, we need two additional variables and  $2|R| + 1$  constraints to represent the Range. In contrast, we need  $|R|^2$  variables and  $2|R|^2$  constraints to represent the Gini deviation (see Table A1 in [Online Appendix A](#)). Thus, the size of the equity-promoting formulation employing the Gini deviation is significantly larger. Such an increase in the size of the IP formulation often suggests an increase in solution time for solving it ([Klotz and Newman, 2013](#)).

Fourth, we observe that Step 3 of the algorithm requires the longest time (see [Online Appendix E](#)) as this step generates the entire set of Pareto optimal solutions by iteratively solving either problem (10) (in Step 3.2) or problem (11) (in Step 3.3). Consider `Inst1-62-A`, for example, the total solution time of this instance using the algorithm with the Range measure is 140 seconds, 133 of which are spent in Step 3. Finally, we observe that instances with vacation requests generated using method B often require a slightly longer time to solve than those with vacation requests generated using method A. This makes sense as method B may lead to more conflicting requests than method A and, thus, potentially harder-to-solve instances.

Note that using the traditional  $\varepsilon$ -constraint method ([Algorithm 1](#)), we could not solve any of the generated RRAP instances under the Gini, SumMaxPair, and MeanDev. Consider `Inst1-125-A` for example. Using the Pareto Search Algorithm ([Algorithm 2](#)), we can obtain all the Pareto optimal solutions (16, 15, 2) with average solution time (1179.2, 1188.7, 1267) seconds for the equity-promoting model with (Gini, MeanDev, SumMaxPair). In contrast, [Algorithm 1](#) cannot find the entire set within a day. In fact, [Algorithm 1](#) terminates at the third iteration with an average MIP gap of (63%, 44%, 79%) for (Gini, MeanDev, SumMaxPair).

Finally, we analyze solution time using formulation (1) and the equity-neutral model (2). In [Online Appendix F](#), Tables F1 and F2 respectively present solution time of formulations (1) and (2) in Tables F1 and F2. Using the equity-neutral model (2), we can solve all instances with an average solution time ranging from 2.5 to 60 seconds. In fact, we can quickly solve even larger (though not realistic) instances of the problem using this model. For example, the average solution time of instances with 200, 400, and 600 residents ranges from 12 to 80 seconds. Similarly, using formulation (1), we can solve all the instances quickly with an average solution time ranging from 0.63 to 7.45 seconds.

The results in this section demonstrate the computational efficiency of our proposed approaches for the RRAP.

## 8. Conclusion

Motivated by our collaboration with CUIMC, we propose and analyze new IP models and approaches for the resident-to-rotation assignment problem (RRAP). First, we derive an IP formulation that finds a feasible rotation and vacation schedule that satisfies all rotation requirements. We show that such a formulation and the corresponding equity-neutral formulation that maximizes the number of satisfied vacation requests lead to disparity in satisfying vacation requests among

residents. To address this, we derive an equity-promoting counterpart, which finds optimal rotation schedules that maximize the total number of satisfied vacation requests and minimize a measure of inequity in satisfied requests among residents. Second, we propose a computationally efficient Pareto Search Algorithm that finds the complete set of Pareto optimal (non-dominated) rotation schedules to the equity-promoting model within a time that is suitable for practical implementation.

Third, we develop a user-friendly tool that implements the proposed methodologies, which helps residency programs automate the rotation schedule. Finally, we construct various instances based on data from CUIMC to derive insights and illustrate the potential benefits of our proposed approaches in practice. Our results demonstrate the computational efficiency and, thus, the potential implementability of our proposed approach. They also show the benefits of our proposed approaches in promoting equity in resident rotation scheduling. Moreover, we demonstrate the superior computational performance of our Pareto Search Algorithm compared with the traditional  $\varepsilon$ -constraint method.

For future research, extending the proposed approaches by considering potential uncertain events that could affect residents' training and rotations (e.g., residents' absenteeism) would be valuable. It will also be interesting to consider residents' preferences on possible shifts within each rotation and build on our proposed approach to promote equity in shift schedules.

## References

- Abdalkareem, Z.A., Amir, A., Al-Betar, M.A., Ekhan, P., Hammouri, A.I., 2021. Healthcare scheduling in optimization context: a review. *Health and Technology* 11, 445–469.
- Akbarzadeh, B., Maenhout, B., 2021. An exact branch-and-price approach for the medical student scheduling problem. *Computers & Operations Research* 129, 105209.
- Ballesteros, E., Romero, C., 2013. Multiple criteria decision making and its applications to economic problems. Springer Science & Business Media.
- Bard, J.F., Shu, Z., Morrice, D.J., Leykum, L.K., 2016. Annual block scheduling for internal medicine residents with 4+ 1 templates. *Journal of the Operational Research Society* 67, 911–927.
- Bard, J.F., Shu, Z., Morrice, D.J., Leykum, L.K., 2017. Constructing block schedules for internal medicine residents. *IISE Transactions on Healthcare Systems Engineering* 7, 1–14.
- Becker, G.E., Wortmann, R., Silva Jr, J., 1982. Computerized house officer schedules at the university of michigan. *Academic Medicine* 57, 308–15.
- Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L., 2013. Personnel scheduling: A literature review. *European journal of operational research* 226, 367–385.

- Bergman, D., Bodur, M., Cardonha, C., Cire, A.A., 2022. Network models for multiobjective discrete optimization. *INFORMS Journal on Computing* 34, 990–1005.
- Block, L., Wu, A.W., Feldman, L., Yeh, H.C., Desai, S.V., 2013. Residency schedule, burnout and patient care among first-year residents. *Postgraduate medical journal* 89, 495–500.
- Brucker, P., Qu, R., Burke, E., 2011. Personnel scheduling: Models and complexity. *European Journal of Operational Research* 210, 467–473.
- Castaño, F., Velasco, N., 2020. Exact and heuristic approaches for the automated design of medical trainees rotation schedules. *Omega* 97, 102107.
- Cire, A.A., Diamant, A., Yunes, T., Carrasco, A., 2019. A network-based formulation for scheduling clinical rotations. *Production and operations management* 28, 1186–1205.
- Cohn, A., Root, S., Kymissis, C., Esses, J., Westmoreland, N., 2009. Scheduling medical residents at boston university school of medicine. *Interfaces* 39, 186–195.
- Colbert, C.Y., French, J.C., Herring, M.E., Dannefer, E.F., 2017. Fairness: the hidden challenge for competency-based postgraduate medical education programs. *Perspectives on Medical Education* 6, 347–355.
- Das, I., Dennis, J.E., 1997. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization* 14, 63–69.
- Ehrgott, M., 2005. *Multicriteria optimization*. volume 491. Springer Science & Business Media.
- Erhard, M., Schoenfelder, J., Fügener, A., Brunner, J.O., 2018. State of the art in physician scheduling. *European Journal of Operational Research* 265, 1–18.
- Franz, L.S., Miller, J.L., 1993. Scheduling medical residents to rotations: solving the large-scale multiperiod staff assignment problem. *Operations Research* 41, 269–279.
- Gross, C.N., Brunner, J.O., Blobner, M., 2019. Hospital physicians can't get no long-term satisfaction—an indicator for fairness in preference fulfillment on duty schedules. *Health Care Management Science* 22, 691–708.
- Guo, J., Morrison, D.R., Jacobson, S.H., Jokela, J.A., 2014. Complexity results for the basic residency scheduling problem. *Journal of Scheduling* 17, 211–223.
- Guo, J., Pozehl, W., Cohn, A., 2023. A two-stage partial fixing approach for solving the residency block scheduling problem. *Health Care Management Science* 26, 363–393.

- Howard, F.M., Gao, C.A., Sankey, C., 2020. Implementation of an automated scheduling tool improves schedule quality and resident satisfaction. *PloS one* 15, e0236952.
- Kim, I.Y., De Weck, O.L., 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization* 29, 149–158.
- Klotz, E., Newman, A.M., 2013. Practical guidelines for solving difficult mixed integer linear programs. *Surveys in Operations Research and Management Science* 18, 18–32.
- Kraul, S., 2020. Annual scheduling for anesthesiology medicine residents in task-related programs with a focus on continuity of care. *Flexible Services and Manufacturing Journal* 32, 181–212.
- Lemay, B., Cohn, A., Epelman, M., Gorga, S., 2016. Scheduling medical residents with conflicting requests for time off. *Scheduling Medical Residents With Conflicting Requests For Time Off*. 61.
- Lemay, B., Cohn, A., Epelman, M., Gorga, S., 2017. New methods for resolving conflicting requests with examples from medical residency scheduling. *Production and Operations Management* 26, 1778–1793.
- Marler, R.T., Arora, J.S., 2010. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization* 41, 853–862.
- Perelstein, E., Rose, A., Hong, Y.C., Cohn, A., Long, M.T., 2016. Automation improves schedule quality and increases scheduling efficiency for residents. *Journal of graduate medical education* 8, 45–49.
- Proano, R.A., Agarwal, A., 2018. Scheduling internal medicine resident rotations to ensure fairness and facilitate continuity of care. *Health Care Management Science* 21, 461–474.
- Shahraki, N., Sir, M.Y., Prindle, T., Ramar, K., 2022. A decision-support system to schedule rotations for trainees. *ATS scholar* 3, 425–432.
- Smalley, H.K., Keskinocak, P., 2016. Automated medical resident rotation and shift scheduling to ensure quality resident education and patient care. *Health care management science* 19, 66–88.
- Uhde, A., Schlicker, N., Wallach, D.P., Hassenzahl, M., 2020. Fairness and decision-making in collaborative shift scheduling systems, in: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

# Equity-promoting Integer Programming Approaches For Medical Resident Rotation Scheduling (Online Appendices)

## Online Appendix A. Equity-promoting RRAP formulation

In this appendix, we present the equity-promoting formulations for the RRAP with each measure listed in Table 3.

### Online Appendix A.1. Equity-promoting RRAP with Range

$$\begin{aligned}
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, u^{\min} \geq 0, u^{\max}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, u^{\min}, u^{\max}}{\text{minimize}} && u^{\max} - u^{\min} \\
 & \text{subject to:} && (1b) - (1m), \quad u^{\max} \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, \forall r \in R, \quad u^{\min} \leq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, \forall r \in R.
 \end{aligned}$$

### Online Appendix A.2. Equity-promoting RRAP with Gini

$$\begin{aligned}
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{minimize}} && \sum_{(r,r') \in R \times R \setminus \{r\}} y_{r,r'} \\
 & \text{subject to:} && (1b) - (1m), \quad y_{r,r'} \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}, \quad \forall (r, r') \in R \times R \setminus \{r\}, \\
 & && y_{r,r'} \geq \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, \quad \forall (r, r') \in R \times R \setminus \{r\}.
 \end{aligned}$$

### Online Appendix A.3. Equity-promoting RRAP with MaxPair

$$\begin{aligned}
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{minimize}} && y \\
 & \text{subject to:} && (1b) - (1m), \quad y \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}, \quad \forall (r, r') \in R \times R \setminus \{r\}, \\
 & && y \geq \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, \quad \forall (r, r') \in R \times R \setminus \{r\}.
 \end{aligned}$$

### Online Appendix A.4. Equity-promoting RRAP with MeanDev

$$\begin{aligned}
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
 & \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{minimize}} && \sum_{r \in R} y_r
 \end{aligned}$$

$$\begin{aligned}
\text{subject to: } & \text{(1b)} - \text{(1m)}, \quad y_r \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \frac{\sum_{r' \in R} \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}}{|R|}, & \forall r \in R, \\
& y_r \geq \frac{\sum_{r' \in R} \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}}{|R|} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, & \forall r \in R.
\end{aligned}$$

Online Appendix A.5. Equity-promoting RRAP with MaxMeanDev

$$\begin{aligned}
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{minimize}} && y \\
\text{subject to: } & \text{(1b)} - \text{(1m)}, \quad y \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \frac{\sum_{r' \in R} \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}}{|R|}, & \forall r \in R, \\
& y \geq \frac{\sum_{r' \in R} \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}}{|R|} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, & \forall r \in R.
\end{aligned}$$

Online Appendix A.6. Equity-promoting RRAP with MaxSumPair

$$\begin{aligned}
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}, \bar{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}, \bar{y}}{\text{minimize}} && \bar{y} \\
\text{subject to: } & \text{(1b)} - \text{(1m)}, \quad y_{r,r'} \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}, & \forall (r, r') \in R \times R \setminus \{r\}, \\
& y_{r,r'} \geq \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, & \forall (r, r') \in R \times R \setminus \{r\}, \\
& \bar{y} \geq \sum_{r' \in R \setminus \{r\}} y_{r,r'}, & \forall r \in R.
\end{aligned}$$

Online Appendix A.7. Equity-promoting RRAP with SumMaxPair

$$\begin{aligned}
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{maximize}} && \sum_{r \in R} \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} \\
& \underset{\mathbf{v}, \mathbf{x}, \mathbf{z}, \mathbf{y}}{\text{minimize}} && \sum_{r \in R} y_r \\
\text{subject to: } & \text{(1b)} - \text{(1m)}, \\
& y_r \geq \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w} - \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w}, & \forall (r, r') \in R \times R \setminus \{r\}, \\
& y_r \geq \sum_{w \in W_{r'}^{\text{vac}}} \sum_{d \in D} v_{r',d,w} - \sum_{w \in W_r^{\text{vac}}} \sum_{d \in D} v_{r,d,w}, & \forall (r, r') \in R \times R \setminus \{r\}.
\end{aligned}$$

Table A1 summarizes the additional number of constraints and variables required to represent



each measure in each equity-promoting RRAP model.

Table A1: Number of additional variables and constraints required to represent each inequity measure.

	Range	Gini	MaxPair	MeanDev	MaxMeanDev	MaxSumPair	SumMaxPair
No. variables	2	$ R ^2$	1	$ R $	1	$ R ^2 + 1$	$ R $
No. constraints	$2 R  + 1$	$2 R ^2$	$ R ^2$	$2 R $	$2 R $	$2 R ^2 +  R $	$2 R ^2$

## Online Appendix B. Mathematical Proofs

### Online Appendix B.1. Proof of Proposition 1

*Proof.* Note that  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0)$  satisfies constraints (1b)-(1m). Hence, it is a feasible solution to problem (4), i.e.,  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}$ . Next, we show that  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}_p$ . To show this, we need to verify that  $f_1(\mathbf{v}) \leq f_1(\mathbf{v}^0)$  and  $f_2(\mathbf{v}^0) \leq f_2(\mathbf{v})$  for any  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$ . Since  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}$ ,  $f(\mathbf{v}^0) = 0$ , and  $f(\mathbf{v}) \geq 0$  for any  $\mathbf{v} \in \mathcal{F}$ , we have  $f_2(\mathbf{v}) \geq f_2(\mathbf{v}^0) = 0$  for all  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$ . Moreover, for any feasible solution  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$  such that  $f_2(\mathbf{v}) = 0$ , we have

$$f_1(\mathbf{v}^0) = \sum_{r \in R} \sum_{b \in B} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w}^0 \geq \sum_{r \in R} \sum_{b \in B} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w} = f_1(\mathbf{v}),$$

where the first inequality follows from the optimality of  $\mathbf{v}^0$ . It follows that  $(\mathbf{z}^0, \mathbf{x}^0, \mathbf{v}^0) \in \mathcal{F}_p$  and  $(f_1(\mathbf{v}^0), f_2(\mathbf{v}^0)) \in \mathcal{P}$ . This completes the proof.  $\square$

### Online Appendix B.2. Proof of Proposition 2

*Proof.* Note that  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l)$  satisfies constraints (1b)-(1m). Hence it is a feasible solution to problem (4), i.e.,  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}$ . Next, we show that  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}_p$ . To show this, we need to verify that  $f_1(\mathbf{v}) \leq f_1(\mathbf{v}^l)$  and  $f_2(\mathbf{v}^l) \leq f_2(\mathbf{v})$ , for any  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$ . Since  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}$ ,  $f_1(\mathbf{v}^l) = \bar{V}$  and  $f_1(\mathbf{v}) \leq \bar{V}$  for any  $\mathbf{v} \in \mathcal{F}$ , we have  $f_1(\mathbf{v}) \leq f_1(\mathbf{v}^l) = \bar{V}$ . Moreover, for any feasible solution  $(\mathbf{z}, \mathbf{x}, \mathbf{v}) \in \mathcal{F}$  such that  $f_1(\mathbf{v}) = \bar{V}$ , we have

$$f_2(\mathbf{v}^l) \leq f_2(\mathbf{v}),$$

where the inequality follows from optimality of  $\mathbf{v}^l$ . Thus,  $(\mathbf{z}^l, \mathbf{x}^l, \mathbf{v}^l) \in \mathcal{F}_p$  and  $(f_1(\mathbf{v}^l), f_2(\mathbf{v}^l)) \in \mathcal{P}$ .  $\square$

### Online Appendix B.3. Proof of Proposition 3

*Proof.* We define  $\mathbf{k} \in \mathbb{R}^{\bar{w}+1}$  as the vacation requests distribution vector associated with vacation schedule  $\mathbf{v}$ , where the  $i$ th entry of  $\mathbf{k} = [k_0, k_1, k_2, \dots, k_{\bar{w}}]$  represents the number of residents with  $i$  satisfied requests and  $\bar{w}$  the is the maximum number of requested weeks for vacation among all residents (i.e.,  $\bar{w} = \max_{r \in R} |W_r^{\text{vac}}|$ ). Note that for each resident  $r \in R$ ,  $u_r$  is the number of residents  $r$ 's satisfied requests ( $u_r = \sum_{d \in D} \sum_{w \in W_r^{\text{vac}}} v_{r,d,w}$ ), and  $\bar{u}$  is the average number of satisfied requests. We assume that there are at least two residents, i.e.,  $|R| \geq 2$ . Next, we show that fulfilling a request for one of the residents with the fewest satisfied requests yields the lowest value of each measure.

**Range.** Let  $l$  and  $s$  represent the largest and smallest non-zero entries of  $\mathbf{k} = [k_0, k_1, k_2, \dots, k_{\bar{w}}]$

(i.e.,  $s \leq u_r \leq l, \forall r \in R$ ). By the definition of the range measure, we have

$$\phi(\mathbf{k}) = \max_{r \in R} u_r - \min_{r \in R} u_r = l - s.$$

Suppose we want to fulfill one additional vacation request and let  $\mathbf{k}' \in \mathbb{R}^{\bar{w}+1}$  represent the resulting vacation request distribution. Consider the following options for fulfilling the additional request.

- a) Fulfill a request of one of the residents with  $i$  granted requests, where  $i = s$ . In this case, if  $k_s \leq 1$ , we find that  $\phi(\mathbf{k}') = l - (s + 1) = l - s - 1$ . Otherwise, if  $k_s \geq 2$ ,  $\phi(\mathbf{k}') = l - s$ .
- b) Fulfill a request of one of the residents with  $i$  granted requests, where  $i \in (s, l)$ . In this case, we observe that  $\mathbf{k}'$ 's largest and smallest non-zero entries  $l$  and  $s$  do not change the inequity value using range does not change (i.e.,  $\phi(\mathbf{k}') = l - s$ ).
- c) Fulfill a request of one resident with  $l$  granted requests, where  $i = l$ . In this case, since we have one resident with  $l + 1$  satisfied requests, the largest non-zero entry of  $\mathbf{k}'$  becomes  $l + 1$ . Thus, the inequity value using range is  $\phi(\mathbf{k}') = l + 1 - s$ .

From (a)–(c), we conclude that fulfilling an additional request of a resident with the fewest satisfied vacation requirements will either decrease the value of the range or decrease it.

**Gini.** Given  $\mathbf{v}$  and its corresponding vacation requests distribution  $\mathbf{k} \in \mathbb{R}^{\bar{w}+1}$ , we compute Gini as

$$\begin{aligned} \phi(\mathbf{v}) &= \sum_{r \in R} \sum_{r' \in R} |u_r - u_{r'}| \\ &= k_0(k_1 + \dots + \bar{w}k_{\bar{w}}) + k_2(2k_0 + k_1 + \dots + (\bar{w} - 1)k_{\bar{w}}) \dots + k_{\bar{w}}(\bar{w}k_0 + (\bar{w} - 1)k_1 + \dots + k_{\bar{w}-1}) \\ &= \sum_{i=0}^{\bar{w}} k_i \left( \sum_{j=0}^{\bar{w}} |j - i| k_j \right), \end{aligned} \tag{14}$$

Let  $p \in [1, \bar{w} - 1]$  and consider the following two options for fulfilling one additional request.

- (a) Fulfill a request of one of the residents with  $p - 1$  granted requests. The new vacation request distribution is  $\mathbf{k}' = [k_0, k_{p-1} - 1, k_p + 1, \dots, k_{\bar{w}}]$  and the Gini deviation  $\phi(\mathbf{k}')$  is as follows.

$$\begin{aligned} \phi(\mathbf{k}') &= \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq [p-1, p]}} |j - i| k_j + |p - 1 - i| (k_{p-1} - 1) + |p - i| (k_p + 1) \right] + (k_{p-1} - 1) \\ &\quad \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p}} |j - (p - 1)| k_j + k_p + 1 \right] + (k_p + 1) \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1}} |j - p| k_j + k_{p-1} - 1 \right] \end{aligned}$$

The difference between  $\phi(\mathbf{k})$  and  $\phi(\mathbf{k}')$  equals to

$$\begin{aligned} \phi(\mathbf{k}') - \phi(\mathbf{k}) &= \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq [p-1, p]}} |j - i| k_j + |p - 1 - i| (k_{p-1} - 1) + |p - i| (k_p + 1) \right] \\ &\quad - \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq [p-1, p]}} |j - i| k_j - |p - 1 - i| k_{p-1} - |p - i| k_p \\ &\quad + k_{p-1} \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p}} |j - (p - 1)| k_j + k_p \right] - (k_{p-1} - 1) \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p}} |j - (p - 1)| k_j + k_p + 1 \right] \end{aligned}$$

$$\begin{aligned}
& + k_p \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1}} |j-p|k_j + k_{p-1} \right] - (k_p + 1) \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1}} |j-p|k_j + k_{p-1} - 1 \right] \\
& = \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left[ |p-i| - |p-1-i| \right] - k_{p-1} + \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p}} |j-(p-1)|k_j + k_p + 1 \right] + k_p \\
& - \left[ \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1}} |j-p|k_j + k_{p-1} - 1 \right] \\
& = \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i + \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1, p}} (|j-(p-1)| - |j-p|)k_j + 2k_p - 2k_{p-1} + 2 \\
& = \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i - \sum_{\substack{j \in [0, \bar{w}]: \\ j \neq p-1, p}} k_j + 2(k_p - k_{p-1} + 1) = 2(k_p - k_{p-1} + 1). \tag{15}
\end{aligned}$$

(b) Fulfill a request of one of the residents with  $p$  granted requests. Let  $\mathbf{k}'' = [k_0, \dots, k_p - 1, k_{p+1} + 1, \dots, k_{\bar{w}}]$  represent the resulting vacation request distribution vector. It is easy to verify following the same steps as in (a) that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) = 2(k_{p+1} - k_p + 1)$ .

Computing the difference between  $\phi(\mathbf{k}') - \phi(\mathbf{k})$  and  $\phi(\mathbf{k}'') - \phi(\mathbf{k})$

$$\begin{aligned}
[\phi(\mathbf{k}'') - \phi(\mathbf{k})] - [\phi(\mathbf{k}') - \phi(\mathbf{k})] &= 2(k_{p+1} - k_p + 1 - k_p + k_{p-1} - 1) \tag{16} \\
&= 2(k_{p+1} + k_{p-1} - 2k_p) \\
&\geq 2[k_{p+1} + k_{p-1} - 2(k_{p+1} + k_{p-1} - |R|)] \\
&\geq 2[2|R| - (k_{p+1} + k_{p-1})] > 0.
\end{aligned}$$

The last inequality holds because  $k_{p+1} + k_p + k_{p-1} \leq \sum_{i=0}^n k_i = |R|$ . Consequently, we conclude that fulfilling a vacation request for one of the residents with  $p \in [1, \bar{w} - 1]$  satisfied requests results in a lower increment in the value of the inequity measure (Gini deviation) than fulfilling a request of one of the residents with  $p + 1$  request. Since we show this for arbitrary  $p$ , this establishes that fulfilling a vacation request for one of the residents with the fewest satisfied requests always yields the lowest value of the Gini deviation.

**Maximum pairwise deviation.** We first show that the value of the maximum pairwise deviation equals that of the range for any vacation request distribution  $\mathbf{k} \in \mathbb{R}^{\bar{w}+1}$ . Recall that  $s$  and  $l$  respectively represent the smallest and largest entries of  $\mathbf{k}$  such that  $k_i > 0, \forall i \in [0, \bar{w}]$ , i.e.,  $s \leq u_r \leq l, \forall r \in R$ . It follows that we can compute the maximum pairwise deviation as

$$\max_{r \in R} \max_{r' \in R} |u_r - u_{r'}| = \max_{r \in R} |u_r - s| = l - s = \max_{r \in R} u_r - \min_{r \in R} u_r.$$

This shows that the range and maximum pairwise deviation are equivalent. It follows from point (1) that the results hold for maximum pairwise deviation.

**Absolute deviation from mean.** By the definition of the absolute deviation from the mean, we have

$$\phi(\mathbf{k}) = \sum_{r \in R} |u_r - \bar{u}| = k_0 \left| \frac{\sum_{j=0}^{\bar{w}} j k_j}{\sum_{j=0}^{\bar{w}} k_j} \right| + \dots + k_{\bar{w}} \left| \bar{w} - \frac{\sum_{j=0}^{\bar{w}} j k_j}{\sum_{j=0}^{\bar{w}} k_j} \right|$$

$$= \sum_{i=0}^{\bar{w}} k_i \left| i - \frac{\sum_{j=0}^{\bar{w}} j k_j}{|R|} \right| = \sum_{i=0}^{\bar{w}} k_i \left| i - \frac{S}{|R|} \right|, \quad (17)$$

where  $S = \sum_{j=0}^{\bar{w}} j k_j$  represents the total number of satisfied vacation requests and  $|R| = \sum_{j=0}^{\bar{w}} k_j$ .

Now suppose we want to fulfill one additional vacation request. Suppose we fulfill a request for one of the residents with currently  $p - 1$  granted requests, where  $p \in [1, \bar{w} - 1]$ . The new vacation request distribution is  $\mathbf{k}' = [k_0, \dots, k_{p-1} - 1, k_p + 1, \dots, k_{\bar{w}}]$ . By equation (17), we have

$$\phi(\mathbf{k}') = \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left| i - \frac{S+1}{|R|} \right| + (k_{p-1} - 1) \left| p-1 - \frac{S+1}{|R|} \right| + (k_p + 1) \left| p - \frac{S+1}{|R|} \right|$$

Computing the difference between  $\phi(\mathbf{k}')$  and  $\phi(\mathbf{k})$ , we obtain

$$\begin{aligned} \phi(\mathbf{k}') - \phi(\mathbf{k}) &= \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left| i - \frac{S+1}{|R|} \right| + (k_{p-1} - 1) \left| p-1 - \frac{S+1}{|R|} \right| + (k_p + 1) \left| p - \frac{S+1}{|R|} \right| \\ &\quad - \left[ \sum_{\substack{i \in [0, \bar{w}]: \\ i \neq p-1, p}} k_i \left| i - \frac{S}{|R|} \right| + (k_{p-1}) \left| p-1 - \frac{S}{|R|} \right| + (k_p) \left| p - \frac{S}{|R|} \right| \right] \\ &= \sum_{i=0}^{\bar{w}} k_i \left[ \left| i - \frac{S+1}{|R|} \right| - \left| i - \frac{S}{|R|} \right| \right] + \left| p - \frac{S+1}{|R|} \right| - \left| p-1 - \frac{S+1}{|R|} \right|. \end{aligned} \quad (18)$$

Now suppose that we instead fulfill an additional request of a resident with  $p$  granted request and obtain the vacation request distribution vector  $\mathbf{k}'' = [k_0, \dots, k_p - 1, k_{p+1} + 1, \dots, k_{\bar{w}}]$ . Using (17) and following the same techniques as in (18), we compute  $\phi(\mathbf{k}'') - \phi(\mathbf{k})$  as

$$\phi(\mathbf{k}'') - \phi(\mathbf{k}) = \sum_{i=0}^{\bar{w}} k_i \left[ \left| i - \frac{S+1}{|R|} \right| - \left| i - \frac{S}{|R|} \right| \right] + \left| p+1 - \frac{S+1}{|R|} \right| - \left| p - \frac{S+1}{|R|} \right|. \quad (19)$$

Subtracting equation (19) from (18), we have

$$\begin{aligned} [\phi(\mathbf{k}'') - \phi(\mathbf{k})] - [\phi(\mathbf{k}') - \phi(\mathbf{k})] &= \left| p+1 - \frac{S+1}{|R|} \right| - \left| p - \frac{S+1}{|R|} \right| - \left| p - \frac{S+1}{|R|} \right| + \left| p-1 - \frac{S+1}{|R|} \right| \\ &= \left| p+1 - \frac{S+1}{|R|} \right| - 2 \left| p - \frac{S+1}{|R|} \right| + \left| p-1 - \frac{S+1}{|R|} \right|. \end{aligned} \quad (20)$$

Equation (20) shows, using the absolute deviation from the mean as the inequity measure, the gap in the inequity value between fulfilling the extra request from a resident with  $p$  granted requests and  $p - 1$  granted requests. Consider the following three cases:

- a)  $p \leq \frac{S+1}{|R|} - 1$  or  $p \geq \frac{S+1}{|R|} + 1$ . We first consider the case when  $p \geq \frac{S+1}{|R|} + 1$ . The same proof techniques can be used for the case when  $p \leq \frac{S+1}{|R|} - 1$ . When  $p+1 > p \geq \frac{S+1}{|R|} + 1$ , we have

$$[\phi(\mathbf{k}'') - \phi(\mathbf{k})] - [\phi(\mathbf{k}') - \phi(\mathbf{k})] = \left( p+1 - \frac{S+1}{|R|} \right) - 2 \left( p - \frac{S+1}{|R|} \right) + \left( p-1 - \frac{S+1}{|R|} \right) = 0.$$

- b)  $\frac{S+1}{|R|} \leq p \leq \frac{S+1}{|R|} + 1$ . In this case,  $p+1 \geq p \geq \frac{S+1}{|R|}$  but  $p-1 \leq \frac{S+1}{|R|}$ . Then we have,

$$\begin{aligned} [\phi(\mathbf{k}'') - \phi(\mathbf{k})] - [\phi(\mathbf{k}') - \phi(\mathbf{k})] &= \left( p+1 - \frac{S+1}{|R|} \right) - 2 \left( p - \frac{S+1}{|R|} \right) + \left( \frac{S+1}{|R|} - p+1 \right) \\ &= 2 \left[ \frac{S+1}{|R|} - p+1 \right] \geq 0. \end{aligned}$$

- c)  $\frac{S+1}{|R|} - 1 \leq p \leq \frac{S+1}{|R|}$ . In this case,  $p+1 \geq \frac{S+1}{|R|}$  but  $p-1 \leq p \leq \frac{S+1}{|R|}$ , then we have

$$[\phi(\mathbf{k}'') - \phi(\mathbf{k})] - [\phi(\mathbf{k}') - \phi(\mathbf{k})] = \left( p+1 - \frac{S+1}{|R|} \right) - 2 \left( \frac{S+1}{|R|} - p \right) + \left( \frac{S+1}{|R|} - p+1 \right)$$

$$= 2 \left[ p - \frac{S+1}{|R|} + 1 \right] \geq 0.$$

From (a)–(c), we conclude that  $[\phi(\mathbf{k}'') - \phi(\mathbf{k})] \geq [\phi(\mathbf{k}') - \phi(\mathbf{k})]$ . It follows that fulfilling a request of a resident with  $p$  requests could result in a larger increment in the value of the inequity measure (absolute deviation from the mean) compared with fulfilling a request to a resident with  $p-1$  requests. Thus, it is optimal to fulfill a request of a resident with the fewest satisfied requests.

**Maximum absolute deviation from mean.** Recall that  $s$  and  $l$  respectively represent the smallest and largest entries of  $\mathbf{k}$  such that  $k_i > 0, \forall i \in [0, \bar{w}]$ , i.e.,  $s \leq u_r \leq l, \forall r \in R$ . By definition of maximum absolute deviation from the mean, we have

$$\begin{aligned} \phi(\mathbf{k}) &= \max_{r \in R} |u_r - \bar{u}| = \max \left\{ |s - \bar{u}|, |s+1 - \bar{u}|, \dots, |l - \bar{u}| \right\} \\ &= \max \left\{ \left| s - \frac{S}{|R|} \right|, \dots, \left| l - \frac{S}{|R|} \right| \right\} \\ &= \max \left\{ \left| s - \frac{S}{|R|} \right|, \left| l - \frac{S}{|R|} \right| \right\}, \end{aligned} \quad (21)$$

where  $S$  is the total number of satisfied requests, i.e.,  $S = \sum_{r \in R} u_r$ . Since the total number of satisfied vacations  $S \in [|R|s, |R|l]$ , the average number of satisfied requests  $\frac{S}{|R|} \in [s, l]$ . Thus, equation (21) reduces to

$$\phi(\mathbf{k}) = \max \left\{ \left( \frac{S}{|R|} - s \right), \left( l - \frac{S}{|R|} \right) \right\}. \quad (22)$$

Suppose we want to fulfill one additional vacation request to a resident and let  $\mathbf{k}'$  represent the resulting new vacation request distribution. Then  $\phi(\mathbf{k}')$  equals

$$\phi(\mathbf{k}') = \max \left\{ \left( \frac{S+1}{|R|} - s' \right), \left( l' - \frac{S+1}{|R|} \right) \right\}, \quad (23)$$

where  $s'$  and  $l'$  represent the smallest and largest non-zero entries of  $\mathbf{k}'$ , respectively. Now, consider the following cases.

a) Fulfill a request of one of the residents with  $s$  granted requests.

First, when  $k_s = 1$  (there is one resident with  $s$  granted requests), we have  $s' = s+1$  and  $l' = l$ . By equation (23),

$$\phi(\mathbf{k}') = \max \left\{ \left( \frac{S+1}{|R|} - (s+1) \right), \left( l - \frac{S+1}{|R|} \right) \right\}, \quad (24)$$

Consider the following three sub-cases:

i)  $\frac{l+s}{2} \leq \frac{S+1}{|R|} - \frac{1}{2}$ . In this case, we have  $l - \frac{S+1}{|R|} \leq \frac{S+1}{|R|} - (s+1)$ . Thus,  $\phi(\mathbf{k}') = \frac{S+1}{|R|} - (s+1)$ . Note also that  $\frac{l+s}{2} \leq \frac{S}{|R|} + \frac{1}{|R|} - \frac{1}{2} \leq \frac{S}{|R|}$  (since  $|R| \geq 2$ ). Hence,  $l - \frac{S}{|R|} \leq \frac{S}{|R|} - s$  and  $\phi(\mathbf{k}) = \left( \frac{S}{|R|} - s \right)$ . The difference between  $\phi(\mathbf{k}')$  and  $\phi(\mathbf{k})$  equals

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = \frac{S+1}{|R|} - s - 1 - \left( \frac{S}{|R|} - s \right) = \frac{1}{|R|} - 1 < 0.$$

ii)  $\frac{l+s}{2} \in \left[ \frac{S+1}{|R|} - \frac{1}{2}, \frac{S}{|R|} \right]$ . Since  $\frac{l+s}{2} \geq \frac{S+1}{|R|} - \frac{1}{2}$ , we have  $l - \frac{S+1}{|R|} \geq \frac{S+1}{|R|} - (s+1)$ . Moreover, since  $\frac{l+s}{2} \leq \frac{S}{|R|}$ , we have  $l - \frac{S}{|R|} \leq \frac{S}{|R|} - s$ . Thus,  $\phi(\mathbf{k}) = \frac{S}{|R|} - s$  and  $\phi(\mathbf{k}') = l - \frac{S+1}{|R|}$ , and

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = l - \frac{S+1}{|R|} - \left( \frac{S}{|R|} - s \right) = l + s - \frac{2S+1}{|R|} \leq \frac{2S}{|R|} - \frac{2S+1}{|R|} < 0.$$

iii)  $\frac{l+s}{2} \geq \frac{S}{|R|}$ . In this case, we have  $l - \frac{S}{|R|} \geq s - \frac{S}{|R|}$  and  $\phi(\mathbf{k}) = \left( l - \frac{S}{|R|} \right)$ . Note that

$l + s \geq \frac{2S}{|R|} \geq \frac{2S}{|R|} + (\frac{1}{|R|} - 1)$  (since  $|R| \geq 2$ ). Thus,  $l - \frac{S+1}{|R|} \geq \frac{S+1}{|R|} - (s + 1)$  and so  $\phi(\mathbf{k}') = (l - \frac{S+1}{|R|})$ . The difference between  $\phi(\mathbf{k}')$  and  $\phi(\mathbf{k})$  equals

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = l - \frac{S+1}{|R|} - l + \frac{S}{|R|} = -\frac{1}{|R|} < 0.$$

We conclude that when  $k_s = 1$ , fulfilling a request of a resident with  $s$  granted request (i.e., the one with the fewest number of satisfied requests) will decrease the value of  $\phi$ .

Now, consider the case when  $k_s \geq 2$  (there are more than two residents with  $s$  granted requests). By equation (23), we have

$$\phi(\mathbf{k}') = \max \left\{ \left( \frac{S+1}{|R|} - s \right), \left( l - \frac{S+1}{|R|} \right) \right\}. \quad (25)$$

Then we consider the following three sub-cases.

- i)  $\frac{l+s}{2} \leq \frac{S}{|R|}$ . In this case,  $l + s \leq \frac{2S}{|R|}$  and  $\frac{S}{|R|} - s \geq l - \frac{S}{|R|}$ . Moreover, since  $\frac{l+s}{2} \leq \frac{S+1}{|R|}$ , we have  $\frac{S+1}{|R|} - s \geq \frac{S+1}{|R|} - l$ . It follows that  $\phi(\mathbf{k}) = \frac{S}{|R|} - s$ ,  $\phi(\mathbf{k}') = \frac{S+1}{|R|} - s$ , and

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = \frac{S+1}{|R|} - s - \frac{S}{|R|} + s = \frac{1}{|R|}.$$

- ii)  $\frac{S}{|R|} \leq \frac{l+s}{2} \leq \frac{S+1}{|R|}$ . In this case, we have  $l - \frac{S}{|R|} \geq \frac{S}{|R|} - s$ . Moreover, since  $\frac{l+s}{2} \leq \frac{S+1}{|R|}$ , by point (i) when  $k_s \geq 2$ ,  $\frac{S+1}{|R|} - s \geq \frac{S+1}{|R|} - l$ . Hence,  $\phi(\mathbf{k}) = l - \frac{S}{|R|}$ ,  $\phi(\mathbf{k}') = \frac{S+1}{|R|} - s$ , and

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = \frac{S+1}{|R|} - s - l + \frac{S}{|R|} = \frac{2S+1}{|R|} - (l + s).$$

Since  $\frac{l+s}{2} \in [S/|R|, (S+1)/|R|]$ ,  $\phi(\mathbf{k}') - \phi(\mathbf{k}) \in [-1/|R|, 1/|R|]$ .

- iii)  $\frac{l+s}{2} \geq \frac{S+1}{|R|}$ . In this case, we have  $l - \frac{S}{|R|} \geq \frac{S}{|R|} - s$  and  $l - \frac{S+1}{|R|} \geq \frac{S+1}{|R|} - s$ . Hence,  $\phi(\mathbf{k}) = l - \frac{S}{|R|}$ ,  $\phi(\mathbf{k}') = l - \frac{S+1}{|R|}$ , and

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = l - \frac{S+1}{|R|} - l + \frac{S}{|R|} = -\frac{1}{|R|}.$$

Consequently, we conclude that when  $k_s \geq 2$ , (i) fulfilling a request to a resident with the fewest satisfied requests will lead to either an increase or decrease in the inequity value; (ii) If the inequity value increases, the increase in the inequity value is at most  $\frac{1}{|R|}$ ; (iii) If the inequity value decreases, the decrease in the inequity value is at most  $\frac{1}{|R|}$ .

- b) Fulfill a request to one of the residents with  $i$  requests, where  $i \in [s+1, l-1]$ . In this case,  $\phi(\mathbf{k})$  equals (24), and the proof is the same as point (a) when  $k_s \geq 2$ . Hence, fulfilling an additional request to a resident with  $[s+1, l-1]$  satisfied requests will lead to either an increase or decrease in the inequity value. The change in the inequity value is at most  $\frac{1}{|R|}$ .

- c) Fulfill a request of one of the residents with  $l$  requests. Let us consider the first case when  $k_l \geq 2$ . In this case,  $\phi(\mathbf{k})$  equals (24). The proof is the same as point (a) when  $k_s \geq 2$ . Now we consider the second case when there is only one resident with  $l$  granted requests ( $k_l = 1$ ). By equation (24), we have

$$\phi(\mathbf{k}') = \max \left\{ \left( \frac{S+1}{|R|} - s \right), \left( l + 1 - \frac{S+1}{|R|} \right) \right\}.$$

Now consider the following three sub-cases.

- i)  $\frac{l+s}{2} \leq \frac{S+1}{|R|} - \frac{1}{2}$ . In this case, since  $l + s \leq \frac{2S+2}{|R|} - 1$ , we have  $(l + 1) - \frac{S+1}{|R|} \leq \frac{S+1}{|R|} - s$ . As in point (a)-(i), we have  $l - \frac{S}{|R|} \leq \frac{S}{|R|} - s$ . Hence,  $\phi(\mathbf{k}) = \frac{S}{|R|} - s$  and  $\phi(\mathbf{k}') = \frac{S+1}{|R|} - s$ ,

Thus,

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = \frac{S+1}{|R|} - s - \frac{S}{|R|} + s = \frac{1}{|R|} > 0.$$

- ii)  $\frac{S+1}{|R|} - \frac{1}{2} \leq \frac{l+s}{2} \leq \frac{S}{|R|}$ . In this case, as in point (a)-ii, we have  $l - \frac{S}{|R|} \leq \frac{S}{|R|} - s$  and  $\phi(\mathbf{k}) = \frac{S}{|R|} - s$ . Moreover, since  $l + s \geq \frac{2S+2}{|R|} - 1$ , we have  $(l+1) - \frac{S+1}{|R|} \geq \frac{S+1}{|R|} - s$  and  $\phi(\mathbf{k}') = l + 1 - \frac{S+1}{|R|}$ . Hence,
- $$\phi(\mathbf{k}') - \phi(\mathbf{k}) = l + 1 - \frac{S+1}{|R|} - \frac{S}{|R|} + s = (l + s + 1) - \frac{2S+1}{|R|} \geq \frac{2S+2}{|R|} - \frac{2S+1}{|R|} > 0.$$
- iii)  $\frac{l+s}{2} \geq \frac{S}{|R|}$ . In this case, as in point (a)-iii, we have  $\phi(\mathbf{k}) = (l - \frac{S}{|R|})$ . Moreover,  $l+1 - \frac{S+1}{|R|} \geq \frac{S+1}{|R|} - s$  and  $\phi(\mathbf{k}') = l + 1 - \frac{S+1}{|R|}$ . Thus,

$$\phi(\mathbf{k}') - \phi(\mathbf{k}) = l + 1 - \frac{S+1}{|R|} - l + \frac{S}{|R|} = 1 - \frac{1}{|R|} \geq 0.$$

Consequently, we conclude that when  $k_l = 1$ , fulfilling an additional request to a resident with the maximal number of satisfied requests will (i) lead to an increase in the inequity value; (ii) an increase in the inequity value is at least  $\frac{1}{|R|}$ . Furthermore, when  $k_l \geq 2$ , fulfilling an additional request to a resident with the maximal number of satisfied requests will lead to either an increase or decrease in the inequity value. The change in the inequity value is at most  $\frac{1}{|R|}$ .

The above analyses show that fulfilling additional request of a resident with the fewest granted requests is optimal because (i) it leads to the maximal decrease in the inequity value, i.e., the decrease in the inequity value is at least  $1/|R|$  when  $k_s = 1$ ; (ii) it leads to a minimal change in the inequity value, i.e., the change in the inequity value is at most  $1/|R|$  when  $k_s \geq 2$ .

**Maximum sum of pairwise deviation.** We first claim that, for any vacation request distribution, the maximum sum of pairwise deviation equals a constant multiplied by the maximum absolute deviation from the mean, i.e.,  $\mathbf{k} \in \mathbb{R}^{\bar{w}+1}$ ,  $\max_{r \in R} \sum_{r' \in R} |u_r - u_{r'}| = |R| \max_{r \in R} |u_r - \bar{u}|$ . To prove this claim, we rewrite the maximum sum of pairwise deviation as

$$\begin{aligned} \phi(\mathbf{k}) &= \max_{r \in R} \sum_{r' \in R} |u_r - u_{r'}| \\ &= \max_{i \in [0, \bar{w}]} \left\{ \sum_{j=0}^{\bar{w}} k_j |j - i| \right\} = \max \left\{ \sum_{i=0}^{\bar{w}} i k_i, \sum_{i=0}^{\bar{w}} (\bar{w} - i) k_i \right\} = \max\{c_1, c_2\}, \end{aligned} \quad (26)$$

where  $c_1$  and  $c_2$  represent the first and second expressions in the max operator, respectively. We first consider the case when  $c_1 \leq c_2$ . In this case, we have  $\phi(\mathbf{k}) = \sum_{i=0}^{\bar{w}} (\bar{w} - i) k_i$  and

$$\begin{aligned} \sum_{i=0}^{\bar{w}} i k_i &\leq \bar{w} \sum_{i=0}^{\bar{w}} k_i - \sum_{i=0}^{\bar{w}} i k_i \\ \frac{\sum_{i=0}^{\bar{w}} i k_i}{\sum_{i=0}^{\bar{w}} k_i} &\leq \frac{\bar{w} \sum_{i=0}^{\bar{w}} k_i - \sum_{i=0}^{\bar{w}} i k_i}{\sum_{i=0}^{\bar{w}} k_i} \\ \bar{u} &\leq \bar{w} - \bar{u}, \end{aligned} \quad (27)$$

where the last inequality follows from  $\bar{u} = \sum_{i=0}^{\bar{w}} ik_i / \sum_{i=0}^{\bar{w}} k_i$ . It follows that when  $c_1 \leq c_2$ , we have

$$\begin{aligned}
\phi(\mathbf{k}) &= \sum_{i=0}^{\bar{w}} (\bar{w} - i)k_i = \sum_{i=0}^{\bar{w}} \bar{w}k_i - \sum_{i=0}^{\bar{w}} ik_i = \sum_{i=0}^{\bar{w}} k_i(\bar{w} - \bar{u}) \\
&= \sum_{i=0}^{\bar{w}} k_i \max\{\bar{u}, |1 - \bar{u}|, \dots, |\bar{w} - \bar{u}|\} \text{ (since } \bar{w} - \bar{u} \geq \bar{u} \text{ by (27))} \\
&= \sum_{i=0}^{\bar{w}} k_i \max_{i \in [0, \bar{w}]} |i - \bar{u}| = |R| \max_{r \in R} |u_r - \bar{u}|.
\end{aligned} \tag{28}$$

From (27), we conclude that  $\max_{r \in R} \sum_{r' \in R} |u_r - u_{r'}| = |R| \max_{r \in R} |u_r - \bar{u}|$ . A similar argument holds when  $c_1 \geq c_2$ . This shows that  $\max_{r \in R} \sum_{r' \in R} |u_r - u_{r'}| = |R| \max_{r \in R} |u_r - \bar{u}|$ . Note that the number of residents  $|R|$  is constant. Moreover, in point (5), we have shown that the proposition holds for the maximum absolute deviation from the mean  $\max_{r \in R} |u_r - \bar{u}|$ . It follows that the results also hold for  $\max_{r \in R} \sum_{r' \in R} |u_r - u_{r'}|$ .

**Sum of maximum pairwise deviation.** Recall that  $s$  and  $l$  represent the smallest and largest entries of  $\mathbf{k}$  such that  $k_i > 0, \forall i \in [0, \bar{w}]$  respectively, i.e.,  $s \leq u_r \leq l, \forall r \in R$ . By definition of the sum of maximum pairwise deviation, we have

$$\begin{aligned}
\phi(\mathbf{k}) &= \sum_{r \in R} \max_{r' \in R} |u_r - u_{r'}| \\
&= k_s(l - s) + k_{s+1} \max\{|(s+1) - s|, |s+2 - (s+1)|, \dots, |l - (s+1)|\} + \dots + k_l(l - s) \\
&= k_s(l - s) + k_{s+1} \max\{(s), (l - s - 1)\} + \dots + k_{l-1} \max\{(l - 1), 1\} + k_l(l - s) \\
&= \sum_{i=s}^l \max\{(i - s), (l - i)\} k_i \\
&= \begin{cases} \sum_{i=s}^{c_1} k_i(l - i) + \sum_{i=c_2}^l k_i(i - s) & \text{If } (l + s)/2 \text{ is fractional} \\ \sum_{i=s}^c k_i(l - i) + \sum_{i=c+1}^l k_i(i - s) & \text{Otherwise} \end{cases}
\end{aligned} \tag{29}$$

where  $c = \frac{l+s}{2}$ ,  $c_1 = \lfloor \frac{l+s}{2} \rfloor$  and  $c_2 = \lceil \frac{l+s}{2} \rceil$ .

In the following, we consider the case when  $(l + s)/2$  is fractional. A similar statement holds when  $(l + s)/2$  is integer. Now suppose we want to fulfill one additional vacation request to a resident with  $p - 1$  granted requests and let  $\mathbf{k}' = [k_0, \dots, k_{p-1} - 1, k_p + 1, \dots, k_{\bar{w}}]$  represent the new vacation request distribution. Consider the following three cases:

a)  $p \leq c_1 - 1$ . In this case, we have  $p - 1 \leq p \leq p + 1 \leq c_1 = \lfloor \frac{l+s}{2} \rfloor$ . Then we know

$$\begin{aligned}
\phi(\mathbf{k}') - \phi(\mathbf{k}) &= \sum_{\substack{i \in [s, c_1]: \\ i \neq p-1, p}} (l - i)k_i + (l - p + 1)(k_{p-1} - 1) + (l - p)(k_p + 1) + \sum_{i=c_2}^l (i - s)k_i \\
&\quad - \left[ \sum_{\substack{i \in [s, c_1]: \\ i \neq p-1, p}} (l - i)k_i + (l - p + 1)(k_{p-1}) + (l - p)(k_p) + \sum_{i=c_2}^l (i - s)k_i \right] \\
&= (l - p + 1)(-1) + (l - p) = l - p - l + p - 1 = -1.
\end{aligned}$$

Now suppose we fulfill an extra request of a resident with  $p$  fulfilled requests and let  $\mathbf{k}''$



be the new vacation request distribution. Since  $p \leq p+1 \leq c_1$ , it is easy to verify that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) = -1$ . Finally, we conclude that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) - [\phi(\mathbf{k}') - \phi(\mathbf{k})] = 0$ .

b)  $p \geq c_2 + 1$ . In this case, we have  $p+1 \geq p \geq p-1 \geq c_2 = \lceil \frac{l+s}{2} \rceil$ . Then we know

$$\begin{aligned} \phi(\mathbf{k}') - \phi(\mathbf{k}) &= \sum_{i=s}^{c_1} (l-i)k_i + \sum_{\substack{i \in [c_2, l]: \\ p-1, p}} (i-s)k_i + (p-1-s)(k_{p-1}-1) + (p-s)(k_p+1) \\ &\quad - \left[ \sum_{i=s}^{c_1} (l-i)k_i + \sum_{\substack{i \in [c_2, l]: \\ p-1, p}} (i-s)k_i + (p-1-s)(k_{p-1}) + (p-s)(k_p) \right] \\ &= (p-1-s)(-1) + (p-s) = p-s-p+1+s = 1 \end{aligned}$$

Since  $p \geq p-1 \geq c_2$ , it is easy to verify that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) = 1$ . Thus, we conclude that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) - [\phi(\mathbf{k}') - \phi(\mathbf{k})] = 0$ .

c)  $c_1 \leq p \leq c_2$ . In this case, since there is no integer between  $c_1$  and  $c_2$ , we have two sub-cases  $p = c_1$  or  $p = c_2$ . Let us consider the first sub-case when  $p = c_1$ . In this case, we have  $p-1 \leq p \leq c_1$ . Then by conclusion in a), we have  $\phi(\mathbf{k}') - \phi(\mathbf{k}) = -1$ . Meanwhile, since  $p+1 \geq c_2 \geq p$ , we have

$$\begin{aligned} \phi(\mathbf{k}'') - \phi(\mathbf{k}) &= \sum_{\substack{i \in [s, c_1]: \\ i \neq p}} (l-i)k_i + (l-p)(k_p-1) + \sum_{\substack{i \in [c_2, l]: \\ i \neq p+1}} (i-s)k_i + (p+1-s)(k_{p+1}+1) \\ &\quad - \left[ \sum_{\substack{i \in [s, c_1]: \\ i \neq p}} (l-i)k_i + (l-p)(k_p) + \sum_{\substack{i \in [c_2, l]: \\ i \neq p+1}} (i-s)k_i + (p+1-s)(k_{p+1}) \right] \\ &= -(l-p) + p+1-s = -l+p+p-s+1 = 2p-(s+l)+1. \end{aligned}$$

Since  $p \geq (l+s)/2 - 1$ , multiplying both sides by two, we have  $2p \geq (l+s) - 2$ . Then we know  $2p - (s+l) + 1 \geq -1$ . Thus we conclude that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) \geq [\phi(\mathbf{k}') - \phi(\mathbf{k})]$ . Now consider the second sub-case when  $p = c_2$ . In this case, we know  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) = 1$  by the conclusion in b). Since  $p \geq c_1 \geq p-1$ , we have

$$\begin{aligned} \phi(\mathbf{k}') - \phi(\mathbf{k}) &= \sum_{\substack{i \in [s, c_1]: \\ i \neq p-1}} (l-i)k_i + (l-p+1)(k_{p-1}-1) + \sum_{\substack{i \in [c_2, l]: \\ i \neq p}} (i-s)k_i + (p-s)(k_p+1) \\ &\quad - \left[ \sum_{\substack{i \in [s, c_1]: \\ i \neq p-1}} (l-i)k_i + (l-p+1)(k_{p-1}) + \sum_{\substack{i \in [c_2, l]: \\ i \neq p}} (i-s)k_i + (p-s)(k_p) \right] \\ &= -(l-p+1) + p-s = -l+p-1+p-s = 2p-(s+l)-1. \end{aligned}$$

Since  $p \leq (l+s)/2 + 1$ , multiplying both sides by two, we have  $2p \leq (l+s) + 2$ , and then  $2p - (s+l) - 1 \leq 1$ . Thus, we conclude that  $\phi(\mathbf{k}'') - \phi(\mathbf{k}) \geq [\phi(\mathbf{k}') - \phi(\mathbf{k})]$ .

The above analyses show that fulfilling a request of a resident with  $p$  requests will result in a larger increment in the value of the inequity value (sum of maximum pairwise deviation) than fulfilling a request of a resident with  $p-1$  requests. It follows that it is optimal to fulfill a request to of resident with the fewest satisfied requests.  $\square$

## Online Appendix C. The Info Template of the RRAP Tool

In this appendix, we provide examples of the input info template of the RRAP tool that the user must fill out and then upload into the interface to generate a rotation schedule. This template is an Excel workbook with seven sheets, each designated for a specific set of input parameters to the IP model. Figures C1–C3 show screenshots of these sheets.

### Input sheet 1: Basic information

Input the name of each resident below, 1 per cell	Input the list of resident classes below, 1 per cell (e.g. PGY1)	Input the list of resident types below, 1 per cell (e.g. Cardiac)	Input the list of services for all classes and types of residents below, 1 per cell	
*max of 100 residents*	*max of 10 classes*	*max of 20 types*	*max of 35 services*	
Resident Names	Classes of Residents	Types of Residents	Services	
Resident A	PGY1	Regular	ACS	
Resident B	PGY2	Cardiac	ACS-OR	
Resident C	PGY3	Prodim	Adult/Anes	
Resident D	PGY4	GU	Allen	
Resident E	PGY5	OMFS	Breast	
Resident F		Ortho	Consults	
Resident G		ENT	CR	
Resident H		Plastics	CTICU	
Resident I			ED	
Resident J			Elective	
Resident K			ENT	
Resident L			HPB	
			HPB-Chabot	
			Interview	
			Lap	
			Nights	

**Step 1:** Input the maximum number of vacations a resident can have in a block

**Step 2:** For each service, input the maximum number of residents on vacation in a week

**Step 3:** Input the services that are in the same department

Department	Service 1	Service 2	Service 3	Max on vacation in a week
1	Rainbow	VTE		1
2	ACS	Consults		1
3				
4				
5				

Services	Max on vacation in a week
ACS-OR	1
Adult/Anes	1
Allen	1
Breast	1
CR	1
CTICU	1
ED	1
Elective	1
ENT	1
HPB	1
HPB-Chabot	1
Interview	1
Lap	1

**Recommendations:**  
Sort services table from A to Z (not required, but easier to fill out future areas)

### Input Sheet 2: Block information

**Step 1:** Input the start date (DD/MM/YYYY) and number of weeks in planning horizon below (cell B7), make sure the start date is on a Monday

**Step 2:** Input the number of weeks in each block, under the Block 1-12 cells. (Other areas will autofill)

**Step 3:** Adjust as necessary if total weeks is red

**Step 4:** Observe the tables below for the date value with each block and week

Number of weeks in model planning horizon	Start Date (Monday)	Number of weeks in given block												Total weeks		
		Resident Class	Number of blocks	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8	Block 9	Block 10		Block 11	Block 12
13	1/1/23	PGY1	12	5	4	5	4	5	4	5	4	5	4	5	4	13
		PGY2	8	7	6	6	7	7	6	7	7	6	7	7	6	53
		PGY3	9	6	6	5	6	6	6	6	6	6	6	6	6	53
		PGY4	7	8	7	7	7	8	7	8	8	8	8	8	8	53
		PGY5	6	9	9	9	9	9	9	9	8					53

Resident Class	Number of blocks	Start Date of a Given Block												Total weeks
		Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8	Block 9	Block 10	Block 11	Block 12	
PGY1	12	1/1/23	2/5/23	3/5/23	4/9/23	5/7/23	6/11/23	7/9/23	8/13/23	9/10/23	10/15/23	11/12/23	12/10/23	13
PGY2	8	1/1/23	2/19/23	4/2/23	5/14/23	7/2/23	8/20/23	10/1/23	11/19/23					53
PGY3	9	1/1/23	2/12/23	3/26/23	4/30/23	6/11/23	7/23/23	9/3/23	10/15/23	11/26/23				53
PGY4	7	1/1/23	2/26/23	4/9/23	6/2/23	7/20/23	9/12/23	11/2/23						53
PGY5	6	1/1/23	3/5/23	5/7/23	7/9/23	9/10/23	11/12/23							53

### Input sheet 3: Residents' impossible blocks and vacation requests

**Step 1:** Assign resident class and type to each resident

**Step 2:** Enter the block numbers that a resident can't be scheduled in (if any) (e.g. If first impossible block a resident has is block 4, input a 4 in the first column of impossible blocks)

**Step 3:** Input each residents vacation requests in date format. (Input the date for the Monday of the week requested)

Resident Names	Resident Class	Resident Type	Impossible Blocks												Step 3: Input each residents vacation requests in date format. (Input the date for the Monday of the week requested)					
			1	2	3	4	5	6	7	8	9	10	11	12	Request Week 1	Request Week 2	Request Week 3	Request Week 4		
Resident A	PGY1	Regular	1	2	3												7/30/23	1/15/23		
Resident B	PGY1	Regular	7	8	9	10											10/8/23	12/17/23		
Resident C	PGY1	Regular															9/17/23	1/22/23		
Resident D	PGY1	Cardiac	4	7	12												8/27/23	3/26/23		
Resident E	PGY1	GU															6/4/23	6/12/23		
Resident F	PGY2	Cardiac	1	2	3	4	5	6									12/24/23	7/9/23		
Resident G	PGY2	Regular															5/14/23	5/28/23		

Refer to 'Num\_Blocks' sheet for dates and corresponding week value for planning horizon

Figure C1: Screenshots of sheets 1–3 of the input info template of the RRAP tool.

### Input sheet 4: Required and impossible departments for each class

**Step 1: Input the different resident classes and their types in the 'Resident Class' and 'Resident Type' columns respectively**  
**Step 2: Input the POSSIBLE (valid) and REQUIRED service minimum values for each class/type combo, IMPOSSIBLE services are represented by a blank cell**

\*max of 40 resident classes/type combinations\*

Resident Class	Resident Type	ACS	ACS-OR	AdultAnes	Allen	Breast	Consults	CR	CTICU	ED	Elective	ENT	HPB	HPB-Chabot
PGY1	Regular	Possible		Required 1x	Possible				Required 1x					Required 1x
PGY1	Cardiac			Required 1x		Required 1x								
PGY1	Prelim	Required 1x												Possible
PGY1	GU	Required 1x		Possible					Required 1x		Required 1x			Possible
PGY1	OMFS		Required 1x	Required 1x									Required 1x	
PGY1	Ortho			Required 2x							Possible			
PGY2	Regular	Required 1x		Required 3x				Required 1x		Required 1x				

### Input sheet 5: The minimum and maximum number of residents in each department

**Input the minimum and maximum values for number of residents in each class for the given department**

\*if class/department combo is impossible, enter 0's for both values (don't leave blanks)

Department	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
ACS	0	2	0	0	0	0	1	1	0	0
ACS-OR	0	0	0	2	0	0	0	0	0	0
AdultAnes	0	1	0	0	0	0	0	0	0	0
Allen	1	3	1	2	0	0	1	1	0	0
Breast	0	3	0	0	0	2	0	0	0	0
Consults	0	0	1	4	0	0	0	0	0	0

### Input sheet 6: Number of required vacations for each class

**Input the residents required number of vacations during the rotations represented here**

Resident Class	Resident Type	Number of Vacations
PGY1	Regular	4
PGY1	Cardiac	3
PGY1	Prelim	4
PGY1	GU	4
PGY1	OMFS	4
PGY1	Ortho	2
PGY1	ENT	1

Figure C2: Screenshots of sheets 4-6 of the input info template of the RRAP tool.

### Input sheet 7: Busy departments

**Indicate the busy departments for a given resident class**

Departments Busy									
Resident Class	Dept1	Dept2	Dept3	Dept4	Dept5	Dept6	Dept7	Dept8	Dept9
PGY1	VTF	Rainbow							
PGY2									
PGY3									
PGY4	Nights								
PGY5									

Figure C3: Screenshots of sheet 7 of the input info template of the RRAP tool.

## Online Appendix D. Additional Details of RRAP Instances

Tables D1, D2, and D3 respectively presents the values of paramaters  $T^{\min}$  and  $T^{\max}$  in Inst1, Inst2 and Inst3. Table D4 provides an example of the values of parameters  $R^{\min}$  and  $R^{\max}$  in Inst1 with 62.

Table D1: The value of  $T^{\min}$  and  $T^{\max}$  for each class of residents in Inst1.

Departments	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	1	1	1	2	0	3	1	4	0	5
Vascular	1	1	0	1	1	0	1	1	0	0
Breast	1	1	0	0	1	1	0	1	0	0
VTF	1	1	0	0	0	2	0	0	0	0
Thoracic	1	1	0	0	1	0	0	0	0	0
CR	1	1	0	0	1	1	0	0	1	0
SICU	1	1	0	0	0	2	0	0	0	1
HPB	1	1	0	0	0	0	1	0	1	0
Peds	1	1	1	0	0	0	0	1	0	1
Rainbow	1	1	0	1	0	0	0	0	0	0
Overlook	1	1	1	0	1	0	1	0	0	0
Lap	1	1	0	1	1	1	0	1	1	0
ACS-OR	0	1	1	0	0	1	0	0	0	1
Consults	0	0	3	1	0	0	0	0	0	0
CTICU	0	0	1	3	0	0	0	0	0	0
Renal	0	0	0	1	1	0	0	0	0	0
Trauma	0	0	0	0	1	1	0	0	0	0
Elective	0	0	0	0	0	2	1	0	1	0
ACS	0	0	0	0	0	0	1	1	0	1
Nights	0	0	0	0	0	0	1	1	1	0
HPB-Chabot	0	0	0	0	0	0	0	1	1	1

Table D2: The value of  $T^{\min}$  and  $T^{\max}$  for each class of residents in Inst2.

Departments	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	1	1	1	1	1	1	1	1	1	1
Vascular	1	1	1	1	1	1	1	1	1	1
Breast	1	1	1	1	1	2	1	1	1	1
VTF	1	1	1	1	1	1	1	1	1	1
Thoracic	1	1	1	1	1	1	1	1	1	1
CR	1	1	1	1	1	2	1	1	1	1
SICU	1	1	1	1	1	1	1	1	1	1
HPB	1	1	1	1	1	1	1	1	1	1
Peds	1	1	1	1	1	1	1	1	1	1
Rainbow	1	1	1	1	1	1	1	1	1	1
Overlook	1	1	1	1	1	1	1	1	1	1
Lap	1	1	1	1	1	1	1	1	1	1

Table D3: The value of  $T^{\min}$  and  $T^{\max}$  for each class of residents in Inst3.

Departments	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	1	1	1	1	1	1	1	1	1	1
Vascular	1	1	1	1	1	1	1	1	1	1
Breast	1	1	1	1	1	1	1	1	1	1
VTF	1	1	1	1	1	1	0	0	0	0
Thoracic	1	1	1	1	1	1	1	1	1	1
CR	1	1	1	1	1	1	1	1	1	1
SICU	1	1	0	0	1	1	1	1	1	1
HPB	1	1	1	1	1	1	1	1	0	0
Peds	1	1	0	0	0	0	0	0	0	0
Rainbow	1	1	0	0	0	0	0	0	0	0
Overlook	1	1	1	1	1	1	1	1	0	0
Lap	1	1	0	0	0	0	1	1	1	1
ACS	0	0	1	1	0	0	0	0	1	1
Consults	0	0	3	3	0	0	0	0	0	0
Trauma	0	0	0	0	1	1	0	0	0	0
Nights	0	0	0	0	0	0	1	1	1	1

Table D4: An example of the value of  $R^{\min}$  and  $R^{\max}$  for each class of residents.

Departments	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	1	3	1	2	0	0	1	1	0	0
Vascular	1	4	0	0	0	0	0	1	0	0
Renal	0	1	0	0	0	1	0	0	0	0
AdultAnes	0	1	0	0	0	0	0	0	0	0
Lap	1	3	0	0	0	1	0	0	0	1
Rainbow	1	3	0	0	0	0	0	0	0	0
ED	0	2	0	0	0	0	0	0	0	0
Breast	0	3	0	0	0	2	0	0	0	0
SICU	1	4	0	0	0	0	0	0	0	0
Plastics	0	2	0	0	0	0	0	0	0	0
Thoracic	1	4	0	0	0	2	0	0	0	0
OMFS	0	1	0	0	0	0	0	0	0	0
PMR	0	1	0	0	0	0	0	0	0	0
HPB	1	3	0	0	0	0	1	1	1	1
ENT	0	1	0	0	0	0	0	0	0	0
Overlook	1	3	0	3	0	2	1	1	0	0
CR	1	3	0	0	0	1	0	0	1	1
ACS	0	2	0	0	0	0	1	1	0	0
VTF	1	3	0	0	0	0	0	0	0	0
Peds	1	4	0	2	0	0	0	0	0	0
PedsAnes	0	1	0	0	0	0	0	0	0	0
ACS-OR	0	0	0	2	0	0	0	0	0	0
Consults	0	0	1	4	0	0	0	0	0	0
Trauma	0	0	0	0	0	1	0	0	0	0
Nights	0	0	0	0	0	0	1	1	1	1
Elective	0	4	0	0	0	0	0	1	0	1
HPB-Chabot	0	0	0	0	0	0	0	0	1	1
CTICU	0	0	1	3	0	0	0	0	0	0

## Online Appendix E. Details of CPU Time of Equity-Promoting model

Table E1 presents solution times in each step of Algorithm 2.

Table E1: Total time (in seconds) in each step of Algorithm 2.

Instance	Range			Gini			MeanDev			MaxMeanDev			SumMaxPair		
	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3
Inst1-62-A	3.6	3.1	133.3	4.6	3.6	197.8	3.9	3.4	185.2	3.9	3.3	141.8	4.5	3.8	176.2
Inst1-62-B	7.7	6.0	194.3	8.4	6.0	337.7	7.0	5.3	431.9	6.9	5.4	209.5	8.0	5.9	291.5
Inst2-62-A	2.5	3.1	110.9	1.5	2.9	114.6	2.2	2.4	108.8	2.2	2.4	75.1	3.1	2.4	113.0
Inst2-62-B	3.5	4.8	135.6	5.4	3.0	122.4	4.1	4.5	108.4	4.1	4.5	76.8	6.4	4.8	93.4
Inst3-62-A	2.3	4.3	103.2	3.1	5.6	171.0	2.5	4.8	153.7	2.5	4.7	117.1	3.1	5.6	146.4
Inst3-62-B	2.2	2.5	149.6	5.0	4.3	636.0	1.9	2.6	66.5	1.9	2.6	61.1	2.4	2.9	182.3
Inst1-125-A	9.8	8.8	539.2	15.5	11.0	1072.4	13.2	9.1	942.9	13.9	9.6	775.4	15.4	11.0	967.8
Inst1-125-B	13.3	9.9	896.8	10.7	8.5	623.9	14.3	9.5	986.4	14.2	9.5	906.1	14.2	9.6	906.1
Inst2-125-A	19.2	22.1	658.0	21.1	21.7	2496.2	18.0	20.9	1438.7	17.9	21.2	1134.2	21.0	22.2	906.2
Inst2-125-B	44.3	28.8	1176.9	43.1	27.9	2931.7	40.2	27.0	1818.6	43.0	27.1	1241.2	43.3	28.2	1101.1
Inst3-125-A	7.9	6.9	510.8	12.0	9.0	1222.5	9.6	7.6	1003.2	9.4	7.5	860.2	11.8	9.0	932.2
Inst3-125-B	14.6	8.0	681.7	14.9	8.5	1201.5	12.8	7.3	956.3	13.7	7.9	818.0	16.8	8.8	855.1
Inst1-185-A	11.2	10.06	1188.3	26	15	3667.1	24.6	12.2	3061.4	23.3	12.3	1915.6	15.4	26.4	950.7
Inst1-185-B	22.6	15.8	1599.2	30.1	18.0	3950.6	30.5	15.7	3563.4	29.1	15.5	2199.8	33.6	21.4	3381.9
Inst2-185-A	20.4	13.1	1316.8	28.7	14.8	5613.6	23.4	12.9	3052.3	23.3	14.7	5248.1	29.2	15.3	2187.7
Inst2-185-B	17.2	8.8	1746.1	20.2	11.6	5668.5	18.2	10.0	3616.5	18.2	9.7	2071.6	21.3	11.9	2697.7
Inst3-185-A	8.5	6.0	903.0	18.5	10.7	3635.5	14.8	8.1	2335.5	14.8	8.1	1789.5	19.0	10.9	2022.0
Inst3-185-B	125.4	59.6	2747.2	115.8	64.8	9718.2	111.5	55.2	7138.0	103.8	36.8	2755.4	116.7	59.6	5423.4

**Online Appendix F. Details of CPU Time of Formulations (1) and (2)**  
 Tables F1 and F2 respectively present solution times of formulations (1) and (2).

Table F1: Solution time (in seconds) of formulation (1).

Instance	Min	Avg	Max
Inst1-62	0.68	0.69	0.70
Inst1-125	2.09	2.12	2.15
Inst1-185	3.38	3.41	3.44
Inst2-62	0.71	0.72	0.73
Inst2-125	2.81	2.87	3.09
Inst2-185	4.42	4.46	4.50
Inst3-62	0.63	0.65	0.66
Inst3-125	3.22	3.26	3.30
Inst3-185	7.37	7.41	7.45

Table F2: Solution time (in seconds) using the equity-neutral model.

Instance	Time	Instance	Time	Instance	Time
Inst1-62-A	3.1	Inst1-125-A	8.8	Inst1-185-A	10.1
Inst1-62-B	6.0	Inst1-125-B	9.9	Inst1-185-B	15.8
Inst2-62-A	3.1	Inst2-125-A	22.1	Inst2-185-A	13.1
Inst2-62-B	4.8	Inst2-125-B	28.8	Inst2-185-B	8.8
Inst3-62-A	4.3	Inst3-125-A	6.9	Inst3-185-A	6.0
Inst3-62-B	2.5	Inst3-125-B	8.0	Inst3-185-B	59.6

**Online Appendix G. Details of the RRAP-Case Instance**

We construct the RRAP-Case instance based on the data related to the CUIMC’s 2023–2024 academic year. As mentioned in Section 7.1, this instance consists of 54 residents, of which (22, 9, 9, 7, 7) are (PGY1, PGY2, PGY3, PGY4, PGY5) residents. There are five types of PGY1 residents, PGY1-Categorical, PGY1-OMFS, PGY1-Prelim, PGY1-GU and PGY1-Ortho. In addition, PGY2 and PGY3 are either Categorical or Cardiac. The number of blocks for (PGY1, PGY2, PGY3, PGY4, PGY5) is (12, 8, 9, 7, 7) and the number of weeks in each block is (4 to 6, 6 to 7, 5 to 6, 7 to 8, 7 to 9). Table G1 summarizes mandatory (required), possible, and busy departments for each type of resident. Mandatory departments are underlined, and busy departments are highlighted in bold text. Table G2 and G3 present the value of parameters  $T^{\min}$  and  $T^{\max}$  for PGY1 and PGY2–PGY5 residents, respectively. Table G4 presents the values of parameters  $R^{\min}$  and  $R^{\max}$ .

Table G1: Mandatory (underlined), possible, and busy departments (bold) for each type of resident.

Resident Type	Departments
PGY1-Categorical	<u>Allen</u> , <u>CR</u> , <u>HPB</u> , <u>Lap</u> , <u>Overlook</u> , <u>Peds</u> , <u>SICU</u> , <u>Thoracic</u> , <u>Vascular</u> , <b><u>Rainbow</u></b> , <b><u>VTF</u></b> , ACS, Breast
PGY1-Prelim	<u>ACS</u> , <u>Breast</u> , <u>Lap</u> , <u>Overlook</u> , <u>Thoracic</u> , <u>Vascular</u> , Peds, HPB, Allen
PGY1-GU	<u>ACS</u> , <u>ED</u> , <u>Vascular</u> , <u>CR</u> , Transplant, <u>SICU</u> , <b><u>Rainbow</u></b> , HPB, Thoracic
PGY1-OMFS	<u>ENT</u> , <u>SICU</u> , <u>AdultAnes</u> , <u>PedsAnes</u> , <u>Plastics</u> , <u>Peds</u> , <u>OMFS</u> , <u>Thoracic</u> , <u>Vascular</u> , <b><u>Rainbow</u></b> , <b><u>VTF</u></b>
PGY1-Ortho	<u>Vascular</u> , <u>Plastics</u> , <u>SICU</u> , <u>PMR</u> , Breast, ED,
PGY2-Categorical	<u>Consults</u> , <u>CTICU</u> , <u>Overlook</u> , <u>Peds</u> , <b><u>ACS</u></b> , <b><u>Allen</u></b>
PGY2-Cardiac	<u>Cardiac</u> , <u>CTICU</u> , <u>Overlook</u> , <u>Consults</u> , <b><u>ACS</u></b> , <b><u>Allen</u></b>
PGY3	<u>Breast</u> , <u>CR</u> , <u>Lap</u> , <u>Overlook</u> , <u>Transplant</u> , <b><u>Thoracic</u></b> , <b><u>Trauma</u></b> , <b><u>Vascular</u></b>
PGY3-Cardiac	<u>Cardiac</u> , <u>Overlook</u> , <u>Breast</u> , <b><u>Thoracic</u></b> , <b><u>Vascular</u></b>
PGY4	<u>ACS</u> , <u>Allen</u> , <u>Elective</u> , <u>HPB</u> , <u>Overlook</u> , <b><u>Nights</u></b> , <b><u>Vascular</u></b>
PGY5	<u>CR</u> , <u>Elective</u> , <u>HPB-Chabot</u> , <u>HPB</u> , <u>Lap</u> , <u>Floats</u> , <b><u>Nights</u></b>

Table G2: The value of  $T^{\min}$  and  $T^{\max}$  for PGY1 residents.

Departments	PGY1-Categorical		PGY1-OMFS		PGY1-Prelim		PGY1-GU		PGY1-Ortho	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	1	1	0	0	0	1	0	0	0	0
CR	1	1	0	0	0	0	1	1	0	0
HPB	1	1	0	0	0	1	0	1	0	0
Lap	1	1	0	0	1	2	0	0	0	0
Rainbow	1	1	1	1	0	0	1	1	0	0
VTF	1	1	1	1	0	0	0	0	0	0
Overlook	1	1	0	0	1	1	0	0	0	0
Peds	1	1	1	1	0	1	0	0	0	0
SICU	1	1	1	1	0	0	1	1	1	1
Thoracic	1	1	1	2	1	2	0	1	0	0
Vascular	1	1	1	1	1	2	1	1	1	1
ACS	0	1	0	0	1	2	1	1	0	0
Breast	0	1	0	0	1	1	0	0	0	1
ENT	0	0	1	1	0	0	0	0	0	0
AdultAnes	0	0	1	1	0	0	0	0	0	0
PedsAnes	0	0	1	1	0	0	0	0	0	0
Plastics	0	0	1	2	0	0	0	0	1	1
OMFS	0	0	1	1	0	0	0	0	0	0
ED	0	0	0	0	0	0	1	1	0	1
Transplant	0	0	0	0	0	0	1	1	0	0
PMR	0	0	0	0	0	0	0	0	1	1

Table G3: The value of  $T^{\min}$  and  $T^{\max}$  for PGY2–PGY5 residents.

Departments	PGY2		PGY2-Cardiac		PGY3		PGY3-Cardiac		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Consults	3	3	1	2	0	0	0	0	0	0	0	0
ACS	1	1	1	1	0	0	0	0	1	1	0	0
Allen	1	1	0	1	0	0	0	0	1	1	0	0
CTICU	1	1	1	1	0	0	0	0	0	0	0	0
Overlook	1	1	1	1	1	1	1	1	1	1	0	0
Peds	1	1	0	0	0	0	0	0	0	0	0	0
Cardiac	0	0	3	3	0	0	4	5	0	0	0	0
Breast	0	0	0	0	1	1	0	1	0	0	0	0
CR	0	0	0	0	1	2	0	0	0	0	1	1
Lap	0	0	0	0	1	2	0	0	0	0	1	1
Transplant	0	0	0	0	1	1	0	0	0	0	0	0
Thoracic	0	0	0	0	1	1	1	2	0	0	0	0
Trauma	0	0	0	0	1	2	0	0	0	0	0	0
Vascular	0	0	0	0	1	1	1	1	1	1	0	0
Elective	0	0	0	0	0	0	0	0	1	1	1	1
HPB	0	0	0	0	0	0	0	0	1	1	1	1
Nights	0	0	0	0	0	0	0	0	1	1	1	1
Chatbot	0	0	0	0	0	0	0	0	0	0	1	1
Floats	0	0	0	0	0	0	0	0	0	0	1	1



Table G4: The value of  $R^{\min}$  and  $R^{\max}$  for each class of residents.

Departments	PGY1		PGY2		PGY3		PGY4		PGY5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Allen	0	1	1	2	0	0	1	1	0	0
Vascular	2	3	0	0	1	1	1	1	0	0
Transplant	0	1	0	0	0	1	0	0	0	0
AdultAnes	0	2	0	0	0	0	0	0	0	0
Lap	1	2	0	0	1	1	0	0	1	1
Rainbow	1	1	0	0	0	0	0	0	0	0
ED	0	2	0	0	0	0	0	0	0	0
Breast	1	2	0	0	0	1	0	0	0	0
SICU	1	3	0	0	0	0	0	0	0	0
Plastics	0	1	0	0	0	0	0	0	0	0
Thoracic	1	2	0	0	1	2	0	0	0	0
OMFS	0	1	0	0	0	0	0	0	0	0
PMR	0	1	0	0	0	0	0	0	0	0
HPB	1	2	0	0	0	0	1	1	1	1
ENT	0	1	0	0	0	0	0	0	0	0
Overlook	0	2	1	2	1	1	0	1	0	0
CR	0	1	0	0	1	2	1	1	1	1
ACS	1	1	1	2	0	0	1	1	0	0
VTF	0	1	0	0	0	0	0	0	0	0
Peds	1	3	0	1	0	0	0	0	0	0
PedsAnes	0	1	0	0	0	0	0	0	0	0
Consults	0	0	2	3	0	0	0	0	0	0
Trauma	0	0	0	0	1	1	0	0	0	0
Nights	0	0	0	0	0	0	1	1	1	1
Elective	0	0	0	0	0	0	1	1	1	1
HPB-Chabot	0	0	0	0	0	0	0	0	1	1
CTICU	0	0	1	2	0	0	0	0	0	0
Floats	0	0	0	0	0	0	0	0	1	1
Cardiac	0	2	0	1	1	1	0	0	0	0